



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Matti Lamppu

Laadunhallinnan kehittäminen elektroniikan valmistuksessa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Electronics

Insinöörityö

25.5.2020

Tekijä Otsikko	Matti Lamppu Laadunhallinnan kehittäminen elektroniikan valmistuksessa
Sivumäärä Aika	34 sivua 25.5.2020
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Ammatillinen pääaine	elektroniikka
Ohjaajat	Osaamisaluepäällikkö Janne Salonen NPI insinööri Eetu Nyman Laatupäällikkö Asko Kataja
<p>Opinnäytetyön tarkoituksena oli uudistaa Kangasalalla toimivan elektroniikan sopimusvalmistajan Jotel Oy:n laadunhallintajärjestelmiä, sekä selkeyttää toimintaa laatua käsittelevissä toimissa. Tätä varten asetettiin neljä tavoitetta: laatudatan keräämisen standardoiminen, tuotannolle paremman kuvan antaminen laadun tilanteesta, korjaustoimenpiteiden priorisointi ja toimenpiteiden vaikutuksien analysointi.</p> <p>Laatudatan kerääminen standardoitiin uudella virhekoodijärjestelmällä, jossa viat luokitellaan ensin tuotantovaiheisiin ja sitten vikatyyppeihin, kuten juotosvikoihin tai komponenttivikoihin. Näiden lisäksi luotiin järjestelmä tuotekohtaisten tyyppivikojen luokitteluun. Vikakoodijärjestelmän viitekehyksenä on käytetty IPC-A-610-standardia, jota yrityksessä käytetään laadun parametrina muutoinkin.</p> <p>Muiden kolmen tavoitteen ratkaisemiseksi kehitettiin ohjelmisto, jolla tuotannon laatudataa voi analysoida, suunnitella analyysien perusteella korjaustoimia, sekä informoida tuotantoa laadun kehityksestä analyyseissä luotujen grafiikkojen avulla. Ohjelmisto käyttää Python-ohjelmointikieltä, ja se on suunniteltu niin, että muutoksien ja uusien analyysien lisääminen on tulevaisuudessa helppoa.</p> <p>Työn tuloksena saatiin aikaan järjestelmä, jossa vikojen kirjaaminen on loogisempaa, sekä työkalu, jonka avulla järjestelmään kerätyn tiedon analysoiminen on helpompaa. Uudistuksen seurauksena on, että tuotannonsuunnittelun ja tuotannon välinen kommunikaatio voidaan suorittaa sujuvammin, kun molemmat osapuolet ovat laadun tilanteesta samalla sivulla.</p>	
Avainsanat	laadunhallinta, elektroniikkavalmistus, taulukot, kuvaajat, ohjelmointi

Author Title	Matti Lamppu Development of Quality Management in Electronics Manufacturing
Number of Pages Date	34 pages 25 th of May, 2020
Degree	Bachelor of Engineering
Degree Programme	Electrical and automation engineering
Professional Major	Electronics
Instructors	Janne Salonen, Head of School, School of ICT, Metropolia Eetu Nyman, NPI Engineer, Jotel Asko Kataja, Quality Manager, Jotel
<p>The purpose of this thesis work was to reform the quality control systems at Jotel Oy, an electronics contract manufacturer operating in Kangasala, and to clarify the procedures that should be used. To this end, four objectives were set: standardizing the collection of product quality data, methods for keeping the production team better informed of product quality situation at present, criteria on how to prioritize remedial actions, and creating better ways to analyze the impact of the remedial measures.</p> <p>The collection of quality data was standardized with a new error code system, in which faults are classified first into production steps and then into fault types such as soldering faults or component faults. In addition to these, a system for classifying product-specific defects was created. The IPC-A-610 standard has been used as the frame of reference for the new error code system, which is already used as a quality parameter at the company.</p> <p>To solve the other three objectives, a software program was developed that can be used to analyze production quality data, plan corrective actions based on the analyses, and inform production about the development of quality using the graphics created in the analyses. The software uses the Python programming language and is designed to make it easy to add changes and new analyses in the future.</p> <p>The result of the thesis work is a new quality control system where recording faults is more logical, as well as a tool that makes it easier to analyze the information collected in the system. The consequence of the reforms is that the communication between production planning and production can be performed more smoothly, as both parties are on the same page about the quality situation.</p>	
Keywords	quality management, electronics manufacturing, spreadsheets, graphs

Sisällys

Käsitteet

1	Johdanto	1
2	Laadunhallinnan nykytilanne	3
2.1	Tuotantoprosessi	3
2.2	Virheiden kirjaaminen	4
2.3	Virheiden korjaus	6
3	Käytettävät teknologiat ja työtä koskevat standardit	7
3.1	Python	7
3.2	Wise	8
3.3	Elektroniikan valmistuksen laatustandardit	9
3.3.1	IPC-A-610-standardi	9
3.3.2	ATEX-, UL- ja CSA-sertifikaatit	9
3.4	Laadunhallinnan standardi ISO-9001:2015	10
4	Laadunvalvonnan kehittäminen	11
4.1	Tietokanta	11
4.2	Vikatietojen kirjaus	11
4.3	Tuotannon lukujen analysointi ja kaaviointi	15
4.3.1	Analyysit	17
4.3.2	Vikojen korjausajan laskeminen	23
4.3.3	Korjausajan arvion analysointi	25
5	Yhteenveto	31
	Lähteet	33

Käsitteet

Aikavaatimus	Aika, jonka jokin tietojenkäsittely vaatii suhteutettuna käsiteltävään tietomäärään.
Keskeinen raja-arvolause	Teoria, jonka mukaan keskiarvo tarpeeksi suurelle joukolle toisistaan riippumattomia satunnaismuuttujia, joilla on jokin määritelty jakauma, on likimain normaalijakauman mukainen jakauma, riippumatta satunnaismuuttujien jakaumasta.
Laadunhallinta	Toimenpiteet, joita jokin organisaatio käyttää sen tarjoamien tuotteiden tai palveluiden hyväksyttävyysskriteerien takaamiseksi, sekä niiden kriteerien valossa tehtävän työn kehitykseen tehokkaasti.
Laaturaportti	Laadunhallintaa varten tehtävä raportti, jonka avulla kerätyllä datalla laadunhallintaa voidaan suorittaa.
Normaalijakauma	Eräs satunnaismuuttujan mittatuloksille käytetty jakauma, tai suuruusjärjestysluokittelu, jota käytetään hyvin usein arvioimaan luonnossa esiintyvien satunnaismuuttujien vaihtelua, esim. aika- tai kokomittauksissa, kun ei todellista jakaumaa tiedetä.
Odotusarvo	Luku, joka kuvaa jonkin satunnaismuuttujan todennäköisintä arvoa.
Poikkeama	Suhteellinen käsitys siitä, kuinka suurelle arvovälille jonkin jakauman arvot hajautuvat.
Satunnaismuuttuja	Jokin mitta, jonka arvo on eri mittauskerralla satunnaisesti erilainen.
Tuotannonohjaus	Menetelmät, joita jokin organisaatio käyttää sen tuotannon laatu-, määrä- ja toimitusaikavaatimusten täyttämiseksi.
Tyypivika	Tietylle tuotteelle ominainen vika.
Varianssi	Luku, joka kuvaa jonkin satunnaismuuttujan hajonnan keskimääräistä vaihtelua sen odotusarvosta.

1 Johdanto

Jotel Oy on suomalainen Kangasalla sijaitseva elektroniikan sopimusvalmistaja, joka valmistaa erilaisia piirilevykokoonpanoja, elektroniikkalaitteita sekä automaatiokeskuk-
sia. Tarkkuutta vaativa käsityö ja tilauksesta valmistettavat pienet ja keskisuuret tuotan-
tosarjat ovat olennainen osa 37 henkilöä vuonna 2018 työllistäneen yrityksen vahvuuk-
sia. [1; 2.]

Opinnäytetyössä on tarkoitus uudistaa yrityksen laadunhallintajärjestelmää. Järjestel-
män uudelleenajattelemisella on yleisellä tasolla tarkoitus selkeyttää toimintaa laadun
ylläpitoa koskevissa asioissa, niin tuotannossa kuin sen suunnittelussa, sekä antaa
konkreettisempaa kuvaa tuotannon laadun todellisesta tilasta.

Uusi järjestelmä otetaan käyttöön ensin yhdessä Jotelin valmistamassa tuoteperheessä.
Tässä yhteydessä järjestelmää kehitetään ja standardoidaan, jonka jälkeen järjestelmä
on helppo mukauttaa erilaisten tuotantoprosessien ja tuotteiden tarpeiden mukaiseksi.

Yrityksessä on myös samaan aikaan meneillään muita tuotannon organisointiin liittyviä
uudistuksia, joten ne vaikuttavat osittain siihen, miten tässä työssä luotellut muutokset
tulevat toimimaan jatkossa. Tarkoitus on kuitenkin, että työssä esitellyt menetelmät laa-
dun parantamiseen tulisivat käyttöön tulevaisuudessa myös yrityksen muissa tuotteissa.

Jotelin valmistamia tuotteita tai konkreettisia tuotannon lukuja, kuten tuotantomääriä,
tuotannossa esiintyvien virheiden oikeita määriä tai myyntilukuja ei käsitellä, sillä ne kuu-
luvat yrityssalaisuuksiin. Tuotannon yksityiskohdista käsitellään vain sellaisia, jotka ovat
oleellisia tämän työn tuomien muutosten osalta. Menetelmien havainnollistamiseen käy-
tetään tuotteille sekä luvuille vastineita, jotka jäljittelevät tuotannossa mahdollisia arvoja.

Työn tavoitteet

Yhdessä Jotelin kanssa olemme asettaneet tälle työlle neljä tavoitetta:

1. laatudatan keräämisen standardoiminen
2. tuotannolle paremman kuvan antaminen laadun tilanteesta
3. korjaustoimenpiteiden priorisointi
4. toimenpiteiden vaikutuksien analysointi.

Laatudatan keräämiseen yrityksessä on tällä hetkellä käytössä tuotannonohjausohjelmisto, johon dataa kerätään tuotannon aikana. Virheiden kirjaus järjestelmässä esitellään tarkemmin luvussa 2.2. Yleisesti listan ensimmäinen tavoite tarkoittaa näiden menetelmien uudelleenajattelua, sekä uusien toimintatapojen ylös kirjausta sekä työntekijöiden niihin kouluttamista.

Tuotannolle parempi kuva laadun tilanteesta saadaan käytännössä siten, että uudesta laatudatasta suoritetaan haluttuja analyyseja, joita voidaan sitten nopealla tahdilla antaa tuotannolle kaavioiden ja graafien muodossa. Ajatuksena on, että kaavioista voisi saada tiedon, esimerkiksi menneiden tuotantoerien hylkäyksien määrästä tai jakauman niissä ilmentyneistä virheistä.

Korjaustoimenpiteiden priorisointia voidaan tehdä, jos tiedetään, kuinka kauan jonkin tietyn tuotteen korjaamiseen kuluu aikaa. Tällä hetkellä tuotannossa korjauksien priorisointi tehdään työntekijöiden kokemuksen pohjalta. Mitä paremmin tiedetään tuotteiden korjausaika, sitä paremmin voidaan tehdä päätöksiä siitä, mitkä viat ovat korjaamisen arvoisia ja mitkä eivät. Mikäli korvaavan osan tekemiseen käytetty aika ja kulut, niin henkilö- kuin materiaalikulut, ovat pienemmät kuin korjauksen käytetty aika ja kulut, ei tuotetta enää kannata korjata. Jos vika on kuitenkin uusi, tarvitaan tietenkin ammattilaisten mielipide, mutta myös tuntemattomalle vialle voidaan laskea keskimääräinen korjausaika.

Korjaustoimenpiteiden priorisointiin voidaan myös laskea vikojen alkuperän parempi jäljittäminen. Jotta vioista voidaan päästä eroon, on olennaista tietää, missä tuotantovaiheessa ne ovat syntyneet, jotta vikojen aiheuttajiin voidaan paremmin puuttua. Läpimenoaikaa ajatellen ei ole mielekästä ryhtyä tarkistamaan tuotteiden toimivuutta jokaisen

tuotantovaiheen jälkeen, joten yleensä perusteellisimmat tarkastukset suoritetaan vasta tuotannon lopussa, jolloin vialliselle tuotteelle on tietysti kertynyt enemmän kuluja. Mitä paremmin vikojen syntyminen voidaan estää, sitä parempi laatu on. Paremman laadun takia kulut laskevat, koska tuotteita ei hylätä niin paljon ja työntekijöille jää enemmän aikaa muihin työtehtäviin.

Toimenpiteiden analysointiin kaivataan myös helpotusta. Kun laatu- ja tuotantotietojen keräämisen uudelleenajattelemisen, tuotannon parempi tiedotus sekä korjaustoimien parempi priorisointi yhdessä parantavat koko laadunkeräysketjun sujuvuutta, voidaan siitä myös paremmin tehdä analyysiä. Analysointia auttavat myös itse tuotannolle tehtävät kaaviot, joista myös tuotannon suunnittelu voi helposti saada kokonaiskuvan laadun kehityksestä.

Jotelin toivomuksesta laadunvalvonnan kehittämisessä käytetään mahdollisimman paljon jo olemassa olevia järjestelmiä, kuten edellä mainittua tuotannonohjausohjelmistoa. Uudistukset yritetään myös toteuttaa siten, että niistä koituisi työntekijöille mahdollisimman vähän ylimääräistä työtä ja että ne olisivat mahdollisimman helppoja käyttää.

2 Laadunhallinnan nykytilanne

Tässä luvussa esitellään tarkemmin laaduntarkastuksen nykytilanne. Luvussa 2.1 esitellään tuotannon prosessi yleisesti, jotta voidaan ymmärtää, miten siinä tarkkailtavat asiat vaikuttavat virheiden kirjaukseen. Luvussa 2.2 käydään läpi virheiden kirjaus tuotannossa käytössä olevassa tuotannonohjausohjelmistossa. Luvussa 2.3 kerrotaan, miten virheiden korjausprosessi tapahtuu.

2.1 Tuotantoprosessi

Tuotanto suoritetaan jokaisella erällä tuotannonohjausohjelmiston avulla tuotannon suunnittelun määrittelemien vaiheiden mukaisesti. Työvaiheet on yleensä jaettu niiden työtehtävien mukaan, esimerkiksi koneladontaan, käsiladontaan tai

loppukokoonpanoon. Työvaiheita voidaan myös lisätä jälkikäteen, jos vaikka halutaan lisätä erillinen korjaustyövaihe. Työvaiheet on suoritettava annetussa järjestyksessä.

Ohjelmistossa on myös merkitty tuotantoerällä käytettävät materiaalit tuotantovaihekohtaisesti. Tuotantoerälle materiaalit saadaan varastosta, josta tuotantoerälle otettava määrä vähentää niiden saldoa. Varaston saldoa tarkkailemalla voidaan tietää, milloin jokin materiaali on loppumassa ja uusi materiaali voidaan tilata ennen kuin tuotanto joudutaan keskeyttämään materiaalin puutoksen takia. Työvaiheelle merkitty materiaali on myös kuitattava käytetyksi, ennen kuin työvaihe voidaan kuitata tehdyksi.

Työntekijät kirjaavat ohjelmassa jokaiselle työvaiheelle työtunteja, jotta tuotannon työvoimakustannukset voidaan laskea ja erotella työvaiheittain. Työtuntien laskeminen tapahtuu siten, että työntekijä kirjaa itsensä sisälle työvaiheelle, ja kirjaa itsensä ulos, kun työvaihe on valmis tai kun työ keskeytetään esimerkiksi lounastauon tai työpäivän päättymisen takia. Järjestelmä laskee näitten kirjausten avulla työtunnit automaattisesti.

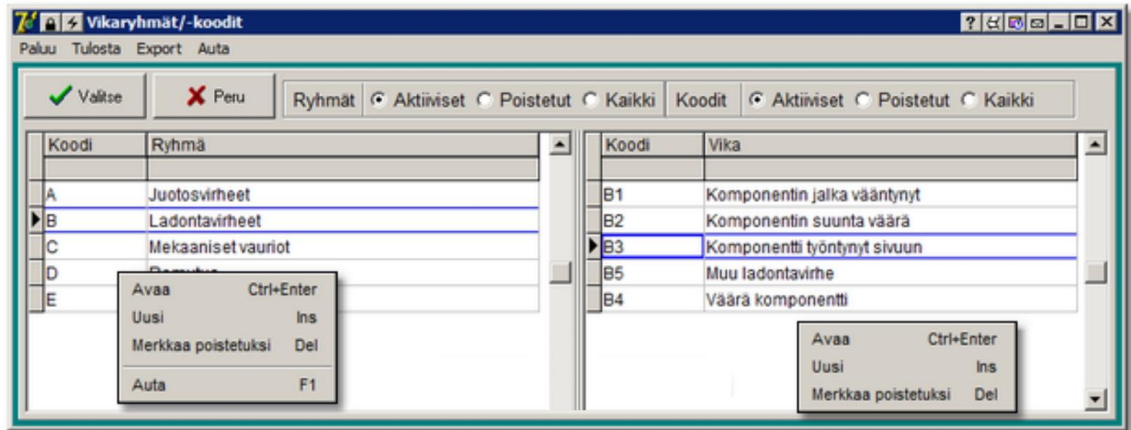
2.2 Virheiden kirjaaminen

Ohjelmistossa virheen kirjauksen voi suorittaa kahdella tavalla:

1. työajan kirjauksen yhteydessä tehtävä luettelointi
2. erikseen kirjoitettava laaturaportti

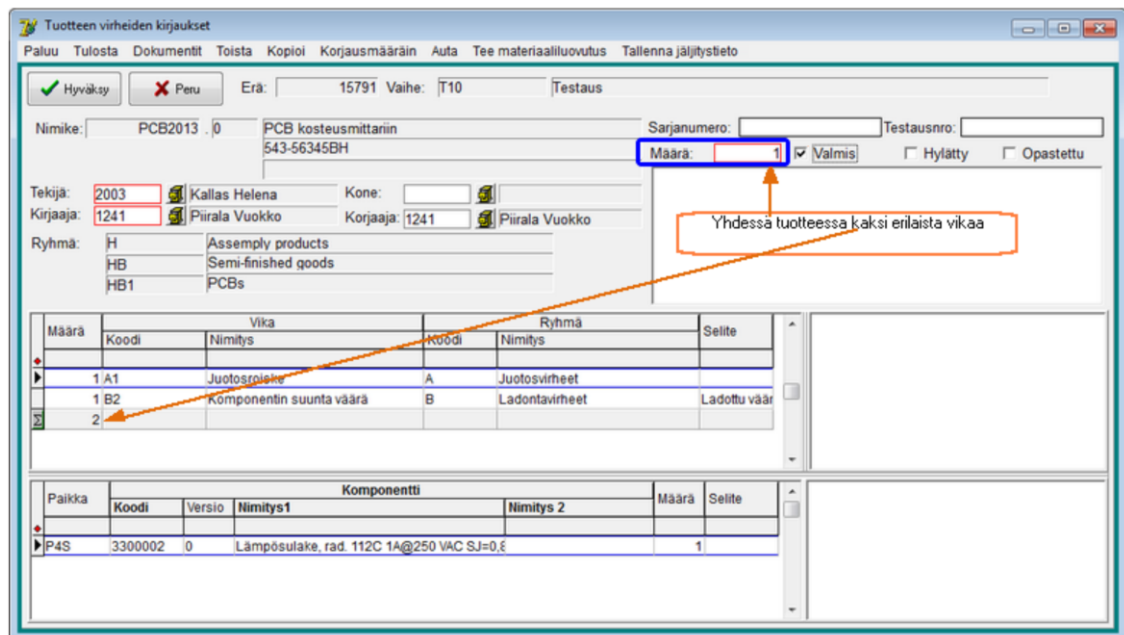
Ensimmäisessä vikakoodit valitaan ennakoon määritellyistä virheistä. Virheet on jaettu ensin erilaisiin vikaluokkiin ja sitten sen vikaluokan erilaisiin vikoihin. Esimerkiksi luokkaan *pintaliitosvirheet* kuuluvat muun muassa virheet *Komponentti väärinpäin*, *Liian vähän tinaa*, *Oikosulku*, jne. Kuva 1 havainnollistaa tätä kirjausjärjestelmää.

Jos virheet kirjataan tällä tavalla, ne merkitään ohjelmistossa erilliseen ”Hylkäysseuranta” taulukkoon, jossa on lueteltu muun muassa, miltä erältä virhe on tullut, kuka sen on merkinnyt, kuinka monta virheistä hylättiin tai korjattiin sekä mikä virhe on kyseessä. Jos virhe merkitään kirjauksen aikana hylätyksi, vähentää kirjaus valmistettavien tuotteiden määrää hylättyjen määrällä.



Kuva 1. Virheiden luettelointi.

Virheet voi kirjata myös tekemällä erillisen laaturaportin (Kuva 2), jossa virheet kohdennetaan tarkemmin johonkin komponenttiin. Jos tätä menettelytapaa käytetään, voidaan vikoihin liittää myös kommentteja, joissa voidaan kuvailla vikaa tarkemmin tai dokumentoida korjausmenetelmät. Tällä tavalla kirjatut virheet merkitään ohjelmistossa "Laadunvalvonta" -taulukkaan, joka on erillinen "Hylkäysseuranta" -taulukosta.



Kuva 2. Laadunvalvontaraportin kirjoittaminen. Raporttiin voidaan liittää useampi niin useampi vika kuin komponentti tuotteeseen käytettyjen materiaalien perusteella. Lisäksi voidaan lisätä teksti useaan eri kohtaan.

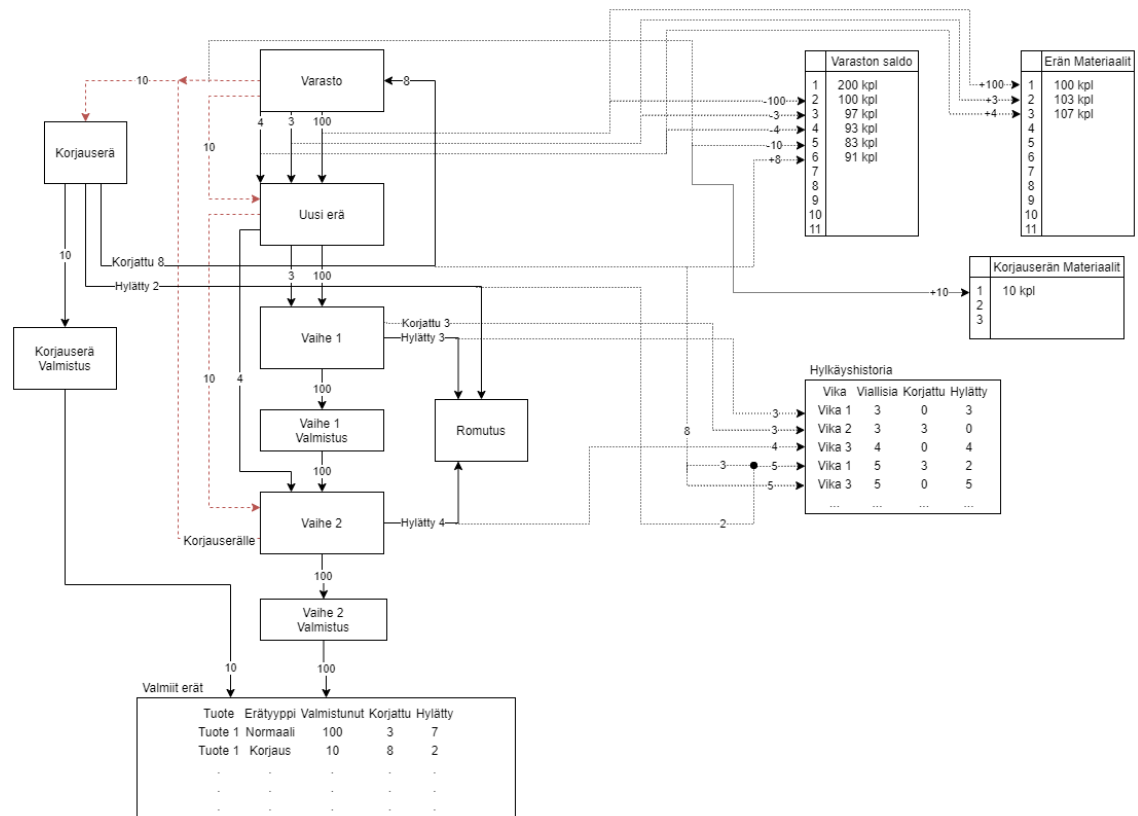
2.3 Virheiden korjaus

Virheet voidaan käsitellä kahdella tapaa; joko erän aikana tai erillisellä korjauserällä. Tuotteesta ja vikojen määrästä riippuen yrityksessä on käytetty molempia tapoja.

Tässä työssä käsiteltävässä tuoteperheessä virheet yritetään käsitellä ensiksi suoraa erän aikana. Mikäli virheitä kertyy useita, tehdään virheiden korjaukselle oma korjaus-työvaihe, johon kirjataan korjaukseen käytetty työaika. Jos tuotteessa olevaa vikaa ei osata suoraa korjata, esimerkiksi sen takia, ettei virhettä pystytä tunnistamaan, tehdään niistä silloin korjauserä. Korjauserälle menevä materiaali korvataan tuotantoerällä varastosta uudella materiaalilla, mikäli sitä on saatavilla. Esimerkiksi rikkiäinen piirilevy saatetaan laittaa korjauserälle ja korvata varastosta uudella levyllä.

Jos korjauksen yhteydessä tarvitsee korvata jokin osa uudella, jos vaikka piirilevyllä oleva rikkiäinen LED vaihdetaan, on korjaukseen käytettävä materiaali lisättävä erälle. Tällä tavalla komponenttien saldot pysyvät kohdillaan. Joitain osia, kuten SMD-vastuksia käsitellään kuitenkin niin suurissa määrissä, ja ne ovat myös niin halpoja, ettei muutaman korvatun osan merkitseminen ole kustannustehokasta.

Kuvassa 3 on havainnollistettu korjausprosessia erän eri vaiheissa. Varastosta erälle lähtevät nuolet arvoilla 3, 4 ja 10 ovat kolmen ja neljän hylätyn sekä 10 korjauserälle lähtevän materiaalin korvaavat materiaalit vastaavasti. Tuotantoerä suoritetaan loppuun ennen korjauserää. Punaisilla nuolilla on merkitty korjauserälle lähtevät tuotteet. Korjauserälle materiaali otetaan ohjelmistossa erälle varastosta, eikä tuotantoerästä, mutta käytännössä uudet tuotteet laitetaan tuotantoerään ja korjattava tuote korjauserään.



Kuva 3. Diagrammi korjausprosessista eri tuotantovaiheissa.

3 Käytettävät teknologiat ja työtä koskevat standardit

Tässä luvussa esitellään laitudatan analysoimiseen käytettävä teknologia peruspiirteis-
sään sekä yrityksessä tässä työssä käsiteltävän tuoteperheen standardit.

3.1 Python

Python on nopean ohjelmoitavuuden sekä valmiiden datankäsittelykirjastojen takia hyvä
valinta tuotannonohjausohjelmistossa kerättävän datan analysoimiseen. Syyt erillisen
datan analysointiin tarkoitetun ohjelman kehittämiseen käsitellään myöhemmin luvussa
4.1. Tässä työssä käytetään Pythonin versiota 2.7, mutta koodi on tehty yhteensopivaksi
versioon 3.8 asti. [3.]

Ohjelmistossa tehtävä datan analysointi suoritetaan käyttäen Pandas-, NumPy- ja Matplotlib-kirjastoja. Pandas on tietojenkäsittelyyn ja analysointiin kehitetty kirjasto, joka soveltuu erityisen hyvin suurien taulukoiden analysointiin sen oman DataFrame-objektin avulla. Pandas tarjoaa myös nopean tavan lukea suuria taulukoita suoraa tiedostosta ja kirjoittaa niitä tiedostoon. [4.] NumPy on tieteelliseen tietojenkäsittelyyn tarkoitettu kirjasto. Sen avulla voidaan käsitellä moniulotteisia taulukoita, sekä suorittaa lineaarialgebran laskutoimituksia, kuten lineaarisia yhtälöryhmiä tai matriiseja. [5.] Matplotlib taas on datan visualisointiin tarkoitettu kirjasto, joka soveltuu hyvin erilaisten graafien ja kaavioiden piirtämiseen. [6.] Nämä kolme kirjastoa ovat osa pythonin tieteellistä jakelupakettia *SciPy Stackiä*, minkä vuoksi ne toimivat hyvin keskenään. [7.]

Ohjelmiston käyttöliittymän tekemiseen on käytetty Tkinter-kirjastoa. Tkinter on Pythonin oletuskirjasto käyttöliittymien tekemiseen. Sen avulla voidaan luoda helposti yksinkertaisia käyttöliittymiä. Peruskirjaston lisäksi käytetään myös tkinter.ttk moduulia sekä Matplotlibin Tkinter rajapintaa siitä saatujen kuvaajien näyttämiseen käyttöliittymässä. [8.]

Näiden kirjastojen lisäksi on käytetty kahta muuta kirjastoa pienemmissä tehtävissä. Nat-sort-kirjastolla voidaan lajitella tuotantojärjestelmän dataa ”luonnollisen lajittelujärjestyksen” mukaan, missä numerot käsitellään atomisesti (jokainen numero erikseen), jolloin lajiteltujen listojen luettavuus paranee. [9.] Cyclor-kirjastoa käytetään yhdessä Matplotlib-kirjaston kanssa laajentamaan graafien väreihin käytettyä värisykliä, eli toistuvaa listaa väreistä, joita graafeihin käytetään. Matplotlib käyttää oletukselta Tableau Classic 10 -värisykliä, mutta se on vaihdettu käyttämään Tableau Classic 20 -värisykliä. [10; 11.]

3.2 Wise

Wise on Oululaisen Wisetime Oy:n kehittämä teollisuusyrityksille suunnattu tuotannon-ohjausjärjestelmä. Järjestelmän avulla voidaan hallita kaikkea yrityksen liiketoimintaprosessien osa-alueita, kuten hankintaa, myyntiä, jäljitystä, sekä dokumenttien-, tuotannon- tai varastonhallintaa. [12.] Wise perustuu Oracle-tietokantaan.

3.3 Elektroniikan valmistuksen laatustandardit

3.3.1 IPC-A-610-standardi

IPC-A-610 on maailmanlaajuinen standardi piirilevyjen kokoonpanon ja muitten elektronisten laitteiden valmistuksen hyväksyttävyysskriteereistä. Standardissa määritellään erilaisten liitoksien, komponenttien ja johtojen valmistuksessa käytettävät minimivaatimukset kuvallisten esimerkkien avulla useassa eri vaativuusluokassa. Standardin tarkoituksena on luoda yhtenäinen käsitys hyväksyttävästä elektroniikan lopputuotteesta sekä taata komponenttien toimivuus ja käyttöturvallisuus. IPC-A-610-standardin oikeaoppiseen omaksumiseen on olemassa koulutusohjelma, jossa henkilö voidaan kouluttaa tuntemaan standardia monella eri tasolla, esimerkiksi kouluttajan tai käyttöönottajan asemasta. IPC-A-610-standardia on käytetty tuotannossa käytettävien vikaluokitusten viitekehyksenä. [13.]

3.3.2 ATEX-, UL- ja CSA-sertifikaatit

ATEX 2014/34/EU -direktiivi on Euroopan unionin määrittelemä säädös räjähdysherkissä tiloissa käytettävien laitteiden suojausvaatimuksista. Suojausvaatimuksilla pyritään estämään laitteiden synnyttämien räjähdysherkkien yhdisteiden tai syttymislähteiden syntymistä tai eristämään ne luotettavasti. Suojausvaatimukset täyttävillä laitteilla on oikeus säädöksen määräämään sertifikaattimerkkiin, jonka avulla laitteen käyttötilat ja turvallisuus on luokiteltu johonkin kolmesta kategoriasta.

Laitteen käyttötilat luokitellaan ATEX-vyöhykkeisiin 0, 1 ja 2 sen mukaan, kuinka todennäköistä räjähdysherkän kaasun ilmentyminen tiloissa on (0 = aina, 1 = mahdollisesti, 2 = epätodennäköisesti), ja samoin vyöhykkeisiin 20, 21 ja 22 sen mukaan, kuinka todennäköistä sama on räjähdysherkälle pölylle tai kuidulle. Laitteen turvallisuus on luokiteltu kategorioihin 1G, 2G, 3G kaasuille ja 1D, 2D, 3D pölylle ja kuidulle sen mukaan, kuinka monta samanaikaista toisistaan riippumatonta vikatilannetta tuotteessa voi esiintyä ilman syttymisvaaraa (1G/D = kaksi vikatilannetta, 2G/D = yksi vikatilanne, 3G/D = vain normaali operaatio taattu). ATEX-direktiivin täyttäminen edellyttää tuotteiden valmistuksessa erityistä huolellisuutta, jonka laadunhallintajärjestelmän tulee mahdollistaa. [14.]

USA:n ja Kanadan laitetestauksen sertifikaattien UL:n ja CSA:n avulla todennetaan tuotteiden täyttävän niitä vastaavien organisaatioiden turvallisuuskriteerit, ja että tuotteen on testannut jokin niiden valtuuttama taho. [15.] Niin USA:n kuin Kanadan lainsäädäntö edellyttää, että verkkovirtaan liitettävien julkisten tai kiinteiden laitteiden tulee täyttää maata vastaava sertifikaatti. [16.]

3.4 Laadunhallinnan standardi ISO-9001:2015

ISO-9001 on kansainvälisesti tunnetuin yritysten laadunhallintaa koskeva standardi. Standardi on tarkoitettu käytettäväksi viitekehyksenä organisaatioiden laadunhallintajärjestelmien kehityksessä, ja sen avulla voidaan jäsentää muun muassa vastuunjakoa, resurssien hallintaa tai toiminnan kehitystä. Standardi perustuu riskilähtöiseen lähestymistapaan, jossa liiketoiminnan riskitekijöitä pyritään hallitsemaan jatkuvasti.

Perusajatuksena laadunhallinnan kehittämiseksi standardissa on jatkuvan parantamisen periaate, niin sanottu PDCA-sykli. Sykli jakautuu neljään vaiheeseen: P = Plan (Suunnittelu), D = Do (Toteutus), C = Check (Tarkistus) ja A = Act (Korjausten teko). Näitä vaiheista seuraamalla yrityksessä on mahdollista vähitellen parantaa haluamaansa liiketoiminnan osa-aluetta havainnoimalla puutteet ja toteuttamalla tarvittavat toimenpiteet.

Tämän työn periaatteena ei ole poistaa yrityksessä tähän malliin kehitettyjä toimintatapoja, vaan selkeyttää ja nopeuttaa niiden toimintaa. Työn tekeminen itsessään voidaan ajatella olevan standardin mukaista toimintaa, joka parantaa toimintatapoja vähitellen. On myös standardin mukaista, että tässä työssä esiteltyjä toimintatapoja kehitetään edelleen niiden käyttöönoton jälkeen. [17.]

4 Laadunvalvonnan kehittäminen

4.1 Tietokanta

Tuotannonohjausjärjestelmästä löytyvillä työkaluilla oli jo entuudestaan mahdollista suorittaa joitain analyyseja paremman tilannekuvan antamiseksi. Nämä työkalut eivät kuitenkaan olleet käyttötarkoituksiimme tarpeeksi laajat, joten päätimme toteuttaa erillisen ohjelmiston, josta halutut analyysit voitaisiin suorittaa. Kehitetty ohjelma esitellään luvussa 4.3.

Ohjelman toimintaa vasten oli selvitettävä tuotannonohjausjärjestelmän tietokantaan pääsy. Järjestelmästä oli jo entuudestaan mahdollista viedä ulos taulukoita halutuun rajaukseen, mutta tämä oli tehtävä manuaalisesti jokaiselle halutulle taulukolle. Koska käyttöön haluttiin tietoa useasta eri taulukosta sekä saada tiedon päivitettyä päivittäin, ei tietojen manuaalinen hakeminen olisi mielekästä pitkällä aikavälillä.

Tämän ratkaisemiseksi otettiin suora yhteys tuotannonohjausohjelmiston kehittäjään Wisetime Oy:hyn. Heidän kanssaan sovittiin ratkaisusta, jossa he kehittäisivät toiminnon edellä mainittujen taulukkojen automaattiseksi ulosvienniksi haluamiemme rajausten perusteella. Tällä tavoin viedyt taulukot korvattaisiin tietyin aikavälein uusilla, josta analyysit sitten voitaisiin tehdä tuotanto-ohjelman ulkopuolella.

4.2 Vikatietojen kirjaus

Tässä työssä käsiteltävän tuoteperheen kohdalla vikojen kirjauksessa yritettiin puuttua erityisesti siihen, että järjestelmästä löytyvät vikakoodit olivat yleensä liian yleispäteviä. Tuotteissa ilmenevät viat ovat yleensä usein toistuvia tyyppivikoja, jolloin ne halutaan yleensä eritellä muista saman tyyppisistä vioista, jotta kyseessä olevan vian kehittymistä voidaan paremmin seurata. Tämän vuoksi tuoteperheessä käytettiin lähinnä erillistä laaturaporttia, mutta sen käytössä oli omat huonot puolensa.

Ensinnäkin vikojen kirjaukseen tällä tavalla kuluu huomattavasti enemmän aikaa, ja toiseksi viat on kuitenkin kirjattava myös luettelosta valittavalla menetelmällä, jotta ohjelmisto merkitsee erälle halutun määrän korjattuja ja hylättyjä tuotteita, ja valmistettavien tuotteiden määrä on näin ollen oikea. Tämän lisäksi raporttiin merkittyjä erällä korjattavia tuotteita ei tarkisteta korjatuiksi ennen erän valmistumista, minkä vuoksi laaturaportti voi olla väärässä, jos sitä ei pidetä yllä huolella.

Tyypivikojen seuraamiseksi uudessa kirjausjärjestelmässä on jokaiselle halutulle tuotteelle oma vikaluokka, joiden alle voidaan luetella täsmällisempiä vikoja. Näiden vikojen määrittäminen ja päivittäminen jää tuotantotiimin vastuulle, sillä he tuntevat eri tuotteiden tyypiviadat parhaiten. Tyypivikojen paikallistaminen ja korjaaminen on tuotteiden laadun kannalta olennaista, sillä ne paljastavat usein tuotannosta systemaattisia epähuomioita, joiden korjaaminen parantaa tuotannon tehokkuutta yleisesti.

Esimerkki: Tuotteessa "Kamera SP4L" on tyypivika "Kameran suljin jumissa". Tämän voisi luokitella pelkäksi komponenttiviaksi, mutta sen erottelu auttaa tätä nimenomaista virhettä poistavien tuotannossa tehtävien muutosten vaikutusten analysointia. Tyypivika merkitään luokkaan SP4L ja sen koodi on esimerkiksi SP4L-1.

Vanhan vikakoodijärjestelmän ideana oli jakaa virheet ensin erilaisiin vikaluokkiin ja sitten niiden alla erilaisiin vikoihin. Vikaluokat pohjautuivat yleisesti johonkin tuotantovaiheeseen, jossa vika on syntynyt. Esimerkiksi pintaliitosvirheet tapahtuvat ladontakoneella ja niihin luokitellut virheet ovat näin ollen ladontakoneella suoritettavissa prosesseissa syntyviä vikoja, kuten itse latomiseen tai komponenttien juottamiseen liittyviä vikoja. Näin ollen pelkän vikaluokan avulla on selvää, missä tuotantovaiheessa vika on syntynyt.

Tuotantovaiheisiin perustuvien vikaluokkien lisäksi vikaluokka saattoi kuvata myös suoraa vian tyyppiä, esimerkiksi *Laiterikko*, *Romutus* tai *Muu vika*. Näiden sisällöt ovat varsin itseselitteisiä, mutta ne eivät kerro vian tuotantovaiheesta mitään, jolloin ne rikkovat edellä mainittua sääntöä, jolla tuotantovaihe voidaan päätellä vikaluokasta.

Uudessa vikaluokituksessa vikaluokat on muutettu kuvastamaan aina suoraa tuotantovaihetta (lukuun ottamatta uusia tuotekohtaisia luokkia), ja muut virheluokat on muutettu virhekategorioksi niiden alle. Virheluokkien ja vikojen koodit on myös muutettu kuvaavimmiksi, esimerkiksi komponentt vikojen vikakoodi alkaa aina "KOM".

Esimerkki: Koneladonnassa tapahtunut ”Puuttuva komponentti” -vika merkitään vikakoodilla ”KON-KOM4”. Jos komponentti taas puuttuu kokoonpanossa, merkitään ”KOK-KOM4”. Näin säilytetään niin tuotantovaiheseurattavuus kuin virheiden kategorisointi niiden tyyppin mukaan.

Taulukoissa 1–7 nimetään uudet vikaluokat ja virheet koodeineen. Uusia tuotekohtaisia tyyppivikoja ja niiden luokkia ei ole luokiteltu tuotesalaisuuksien nojalla. Viat pohjautuvat edellä käsitellyn IPC-A-610-standardin määrittelemiін kelpoisuuskriteereihin.

Taulukko 1. Tuotantovaiheista johdetut virheluokat ja niiden koodit.

Tuotantovaiheet	Virheluokkakoodi
Vastaanotto	VAS
Koneladonta	KON
Käsinladonta	KAS
Aaltojuotos	AAL
Kokoonpano	KOK
Testaus	TES
Korjaus	KOR

Taulukko 2. Komponenttivilat ja niiden koodit.

Vikakoodi	Vika Nimi	Globaali nimi
KOM1	Komponentti vinossa	Component Misaligned
KOM2	Komponentti väärinpäin	Wrong Orientation of Component
KOM3	Komponentin jalka ilmassa	Lifted Component Lead
KOM4	Puuttuva komponentti	Missing Component
KOM5	Väärä komponentti	Wrong Component
KOM6	Komponentin mekaaninen vika	Component Mechanical Failure
KOM7	Komponentin sähköinen vika	Component Electrical Failure
KOM8	Komponentin visuaalinen vika	Component Visual Failure

Taulukko 3. Juotosvirheet ja niiden koodit.

Vikakoodi	Vika Nimi	Globaali nimi
JUO1	Kylmäjuotos	Cold Solder Joint
JUO2	Palanut juotos	Overheated Solder Joint
JUO3	Liian vähän tinaa	Solder Starved Joint
JUO4	Liikaa tinaa	Too Much Solder
JUO5	Oikosulku	Solder Bridging
JUO6	Puuttuva juotos	Missed Solder Joint
JUO7	Huono juotoskontakti	Insufficient Wetting
JUO8	Tinaa roiskunut	Stray Solder Splatter
JUO9	Vääräntyyppinen juotosmateriaali	Wrong Type of Solder

Taulukko 4. Piirilevyn viat ja niiden koodit.

Vikakoodi	Vika Nimi	Globaali nimi
PII1	Folion virhe	Copper Foil Damaged
PII2	Juotosmaskin virhe	Solder Mask Failure
PII3	Porausvirhe	Drilling or Routing Failure
PII4	Vaurioitunut levy	Board Damaged

Taulukko 5. Laiterikkoviat ja niiden koodit.

Vikakoodi	Vika Nimi	Globaali nimi
LAI1	Ladontakone rikki	Pick & Place Machine Broken
LAI2	Reflow-uuni rikki	Reflow Oven Broken
LAI3	Pastakone rikki	Solder Paste Printer Broken
LAI4	Aaltojuotoskone rikki	Wave Soldering Machine Broken
LAI5	Ohjelmointilaite rikki	Device Programmer Broken
LAI6	Testeri rikki	Testing Machine Broken

Taulukko 6. Lopputuoteviat ja niiden koodit.

Vikakoodi	Vika Nimi	Globaali nimi
LOP1	Tuote ei läpäise testiä	Product Does Not Pass Test
LOP2	Tuotteen visuaalinen virhe	Product Visual Failure
LOP3	Tuotteesta puuttuu osia	Product Missing Parts

Taulukko 7. Muut viat ja niiden koodit.

Vikakoodi	Vika Nimi	Globaali nimi
MUU1	Virhe dokumenteissa	Documentation Error
MUU2	Softavika	Software Error
MUU3	Siirretty korjauserälle	Moved to Repair Batch

4.3 Tuotannon lukujen analysointi ja kaaviointi

Tuotannon lukujen analysointiin kehitettiin ohjelma, jossa on mahdollista suorittaa haluttuja analyysseja tuotannonohjausjärjestelmästä haetusta datasta. Ohjelmisto toteutettiin python-ohjelmointikielellä, ja siinä käytetyt kirjastot on esitelty luvussa 3.2.

Ohjelmiston käyttöliittymässä on kaksi näkymää. Ensimmäisessä näkymässä (Kuva 4) voidaan suorittaa haluttuja analyysseja hakemalla joko tuotteita, virheitä tai tuotantovaiheita hakukentässä, kun oikea toiminto on valittuna. Tämän jälkeen voidaan valita toiminnosta riippuen joko haluttu määrä viimeisiä eriä tai päiviä ja luoda määritelty kuvaaja kenttään. Näin voidaan suorittaa nopeita kerta-analyysseja.

Toisessa näkymässä (Kuva 5) analyysseja voidaan tallentaa listaan, josta niitä suoritetaan automaattisesti halutuun aikaväleihin. Analyysien ajoitettu suorittaminen toimii ajoittamalla ohjelman python-tiedoston suorituksen Windowsin Tehtävien ajoituksen avulla.

Kuva 4. Kerta-analyysien tekoon käytetty näkymä. Toiminto valitaan kentän kohdalta avautuvasta pudotusvalikosta. Hakukenttään avautuu pudotusvalikko, kun siihen kirjoitetaan ja se täyttää hakuvalikoita kirjoitettaessa.

Kuva 5. Automaattisesti suoritettavien analyysien tekoon käytettävä näkymä. Taulukko voidaan lajitella painamalla otsikoita, siitä voidaan kopioida kohteita hakukenttään tuplaklikkaamalla ja poistaa kohteita *Delete*-nappaimella. Hakukohteet säilyvät siirryttäessä näkymästä toiseen.

4.3.1 Analyysit

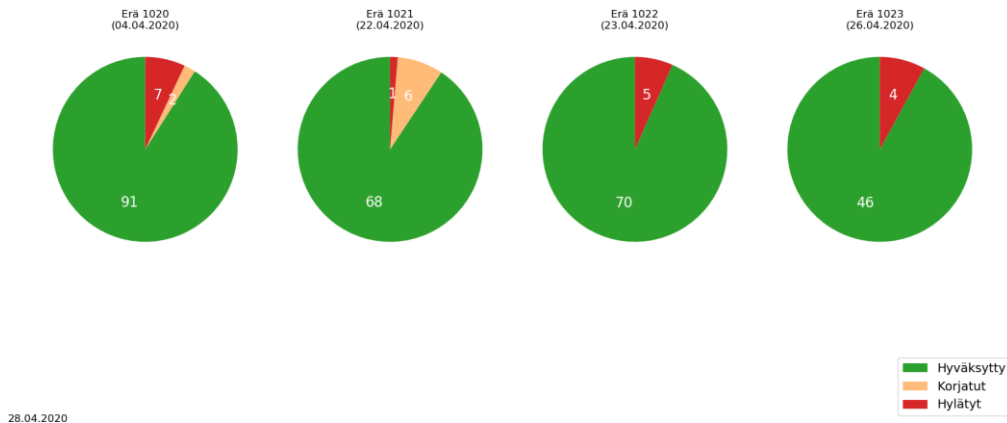
Mahdollisia analyyseja ohjelmistossa karkeasti seitsemän.

1. Erien onnistuneisuus
2. Erien virheiden luokittelu
3. Odottavat korjauserät
4. Työvaiheiden virhejakauma
5. Vikojen korjausaika
6. Vian korjauksen onnistuneisuus
7. Ajankäyttö tuotteelle

Erien onnistuneisuuden analyysin voi suorittaa joko tuotantoerille tai korjauserille. Sen voi myös suorittaa joko halutulle määrälle edellisiä eriä tai päiviä. Valinnat voidaan tehdä valitsemalla niitä vastaava toiminto, joko *Tuotantoerien onnistuneisuus*, *Korjauserien onnistuneisuus*, *Tuotantoerien onnistuneisuus aikavälillä* tai *Korjauserien onnistuneisuus aikavälillä* (samantyyppistä erottelua on käytetty myös muissa analyyseissä). Jos analyysi suoritetaan jollekin määrälle edellisiä eriä, on kuvaaja halutun kokoinen joukko ympyrädiagrammeja, joissa jokainen ympyrä kuvaa yhtä erää. Jos analyysi suoritetaan jollakin aikavälillä, on kuvaaja vain yksi ympyrädiagrammi, joka kuvaa sillä aikavälillä valmistuneiden erien yhteistä onnistuneisuutta.

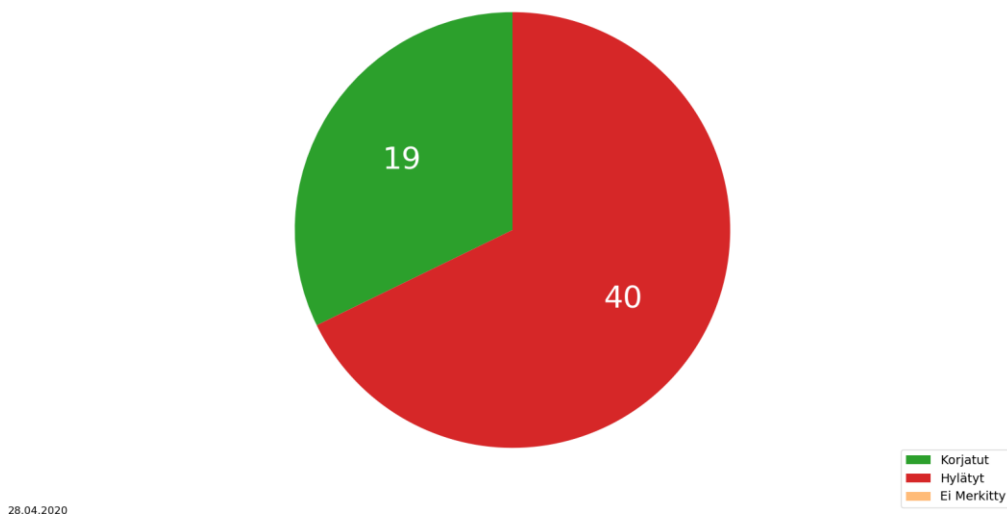
Erien onnistumisella voidaan havainnollistaa erillä tapahtuneiden hylkäyksien ja korjauksien määrää ja verrata sitä läpimenneisiin. Jos näitä lukuja käsitellään prosentteina, puhutaan tuotteiden saannosta, tai suorasaannosta, jos mukaan lasketaan myös korjatut tuotteet. Analyysi näyttää luvut kappalemäärinä, sillä tuotantoerät saattavat olla hyvin erikokoisia, jolloin tuotantoerän määrä pitäisi mainita jokaisen diagrammin kohdalla erikseen, jotta prosenttiosuus antaisi oikean tiedon kappalemääristä. Itseasiassa ympyrädiagrammin palasista saa jo arvion prosenttiosuudesta, jolloin kappalemääriä käyttämällä saadaan tuotteiden määrä sekä hylkäyksien ja korjauksien osuus esitettyä selkeästi.

Tuotantoerien onnistuneisuus tuotteelle Tuote2 viimeisessä 4 erässä



Kuva 6. Erän onnistuneisuus neljälle viimeiselle tuotantoerälle. Kehitystä tälle tuotteelle ei ole tapahtunut, vaan saantoprosentti on pysynyt suunnilleen samana.

Tuotteen Tuote2 korjauksien onnistuneisuus viimeisen 60 päivän aikana

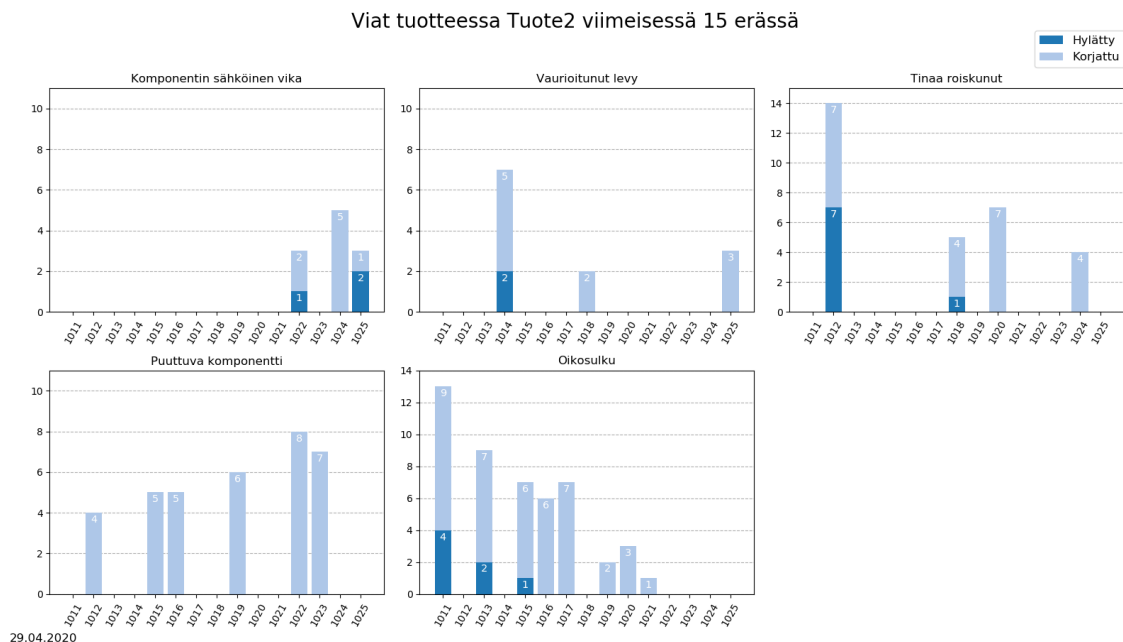


Kuva 7. Erien onnistuneisuus viimeisen 60 päivän ajalta. Tälle tuotteelle on selvästi tehtävä parannuksia virheiden korjauksessa.

Tarkastelemalla erien onnistuneisuutta voidaan saada käsitys virheiden määrän kehityksestä. Virheiden määrästä voidaan suoraan arvioida tuotteille kertyneet lisäkulut tai niiden tuotannossa menetetty arvo. Mikäli tarkastellaan korjauserien onnistuneisuutta, voidaan puhua myös siitä, miten hyvin tuotteita onnistutaan korjaamaan, mikä kertoo korjauksien kannattavuudesta. Kuvaajassa ei ole eroteltu virheitä, sillä se tekisi kuvaajasta

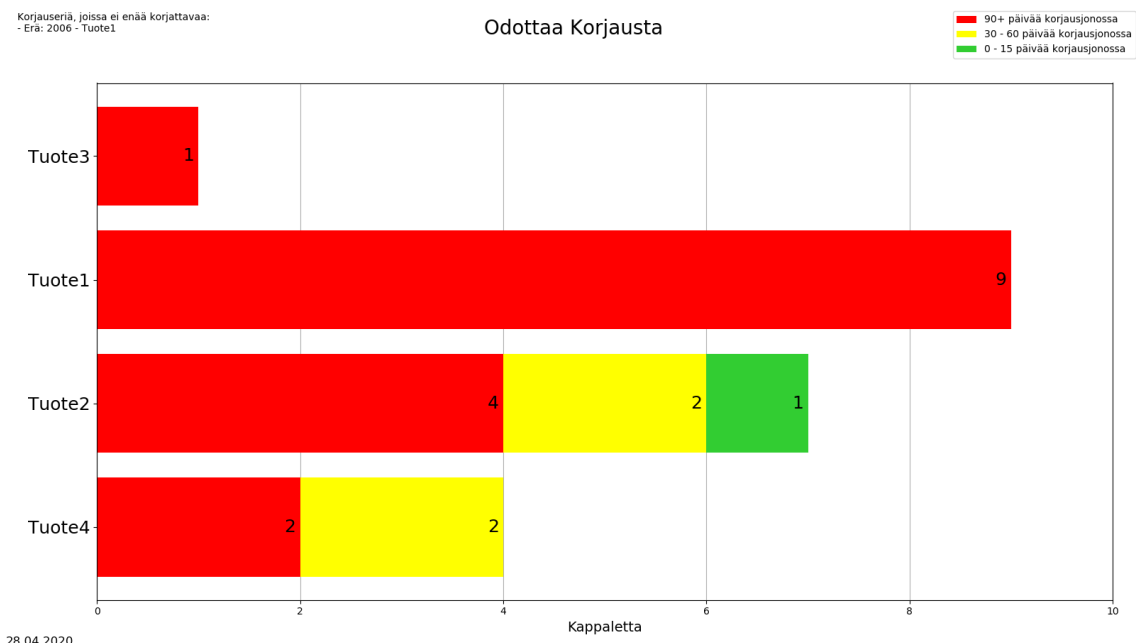
vaikeaselkoisen, varsinkin jos erityyppisiä virheitä on paljon, vaan virheiden erottelu voidaan tehdä omalla toiminnollaan.

Virheiden erottelussa (Kuva 8) luodaan dynaamisesti joukko pylväsdiagrammeja riippuen annetulla erävälillä olevien erilaisten virheiden määrästä. Jokaisessa pylväsdiagrammissa on eroteltu hylätyt ja korjatut virheet. Näin voidaan tarkastella eri virheiden määrän sekä korjattujen ja hylättyjen suhteen yksittäistä kehitystä.

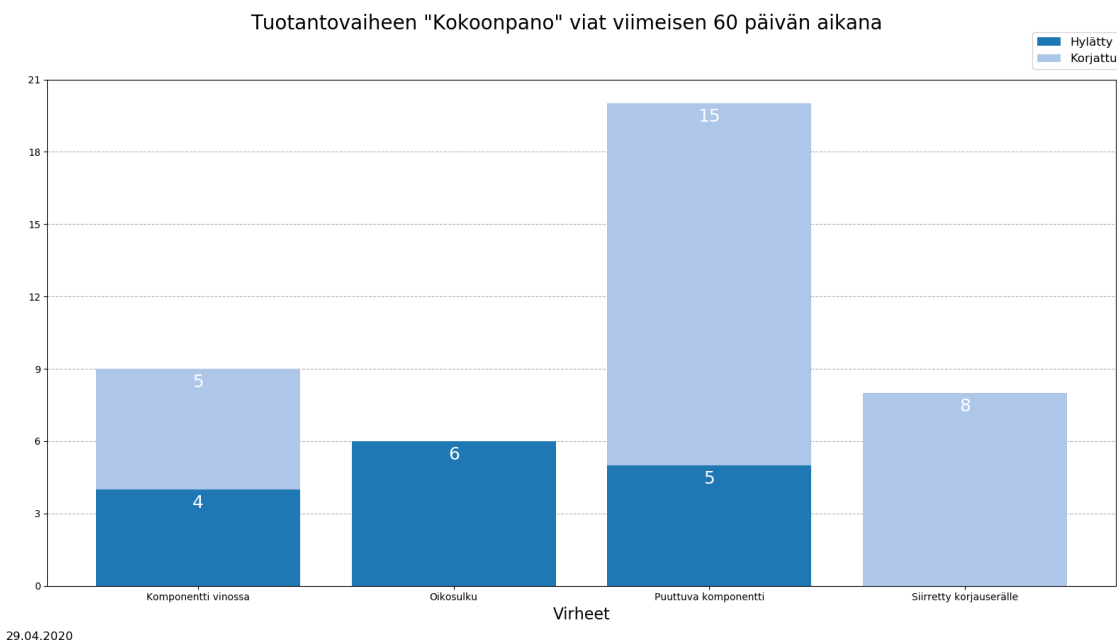


Kuva 8. Virheiden erittely viimeiselle 15 tuotantoerälle. Virhe *Oikosulku* on kuvan mukaan saatu poistettua erien edetessä, kun taas virhe *Puuttuva komponentti* on kasvussa. Viimeisissä erissä on myös ilmentynyt muutama *Komponentin sähköinen vika*.

Odottavat korjauserät analyysissä (Kuva 9) luodaan kuvaaja, jossa avoimilla korjauserillä olevat tuotteet on eroteltu pylväisiin ja pylväät on sitten jaettu palasiin riippuen siitä, kuinka kauan tuotteen eri erät ovat olleet korjausjonossa. Näin voidaan priorisoida kauan korjausta odottaneita tuotteita tai odottaa tuotteiden korjausta, kunnes tietty määrä tuotteita voidaan korjata yhtä aikaa. Vasempaan nurkkaan ilmestyy ilmoitus, mikäli havaitaan avoin korjauserä, jolla ei ole enää korjattavaa, eli korjauserä on jäänyt merkitsemättä valmiiksi.



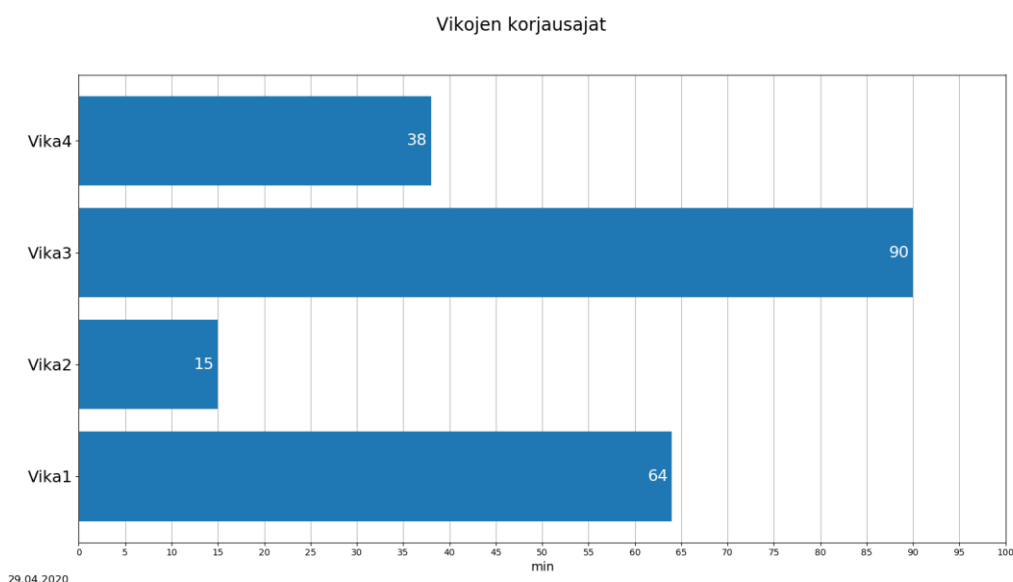
Kuva 9. Odottavien korjauserien kuvaaja. Tuotetta 1 on korjauksessa selvästi eniten, joten sen korjaaminen on luultavasti kustannustehokkainta. Tuotetta 2 on sen sijaan tullut pitkällä välillä useampana kertana. Toisaalta tuotteelle 3 on vain yksi korjattava, joten se saataisiin pois korjauksesta, jos tämä yksi korjattaisiin.



Kuva 10. Virhejakauma työvaiheelle "Kokoonpano". Tämän tuotantovaiheen suurin ongelma on tällä hetkellä joko *Puuttuva komponentti* tai *Oikosulku*. Ensimmäistä on selvästi enemmän, mutta se on saatu yleensä korjattua, mutta toinen on aiheuttanut enemmän hyläyksiä.

Työvaiheiden virhejakaumalla (Kuva 10) voidaan erotella jollain työvaiheella tapahtuneet virheet. Virheet on lisäksi eroteltu hylättyihin ja korjattuihin virheisiin. Näin voidaan tarkastella, mikä niistä on suurin ongelma jollakin työvaiheella. Virhejakauman kehitystä voidaan seurata ajoittamalla analyysi ja vertailemalla lukuja vanhoihin kuvaajiin.

Jotta vikojen korjauksia voidaan priorisoida, on tiedettävä, kuinka kauan niiden korjaamiseen kuluu aikaa. Jokainen tuote on korjattuna yhtä arvokas, joten nopeinten korjatulla tuotteella on näin ollen pienimmät korjauskulut. Toisaalta voidaan myös arvioida jonkin vian korjauksen kustannustehokkuus, jos korjaukseen käytettävä materiaali voidaan vian kuvauksesta arvioida. Esimerkiksi 50 € tuotteeseen tehtävä 20 € korjaus ei välttämättä ole kustannustehokasta, varsinkaan jos korjausaika on pitkä. Työntekijät voivat arvioida korjausajan kokemuksensa pohjalta, mutta mitä tarkempaa tietoa osataan antaa, sen paremmin voidaan priorisoida oikeita korjauksia.

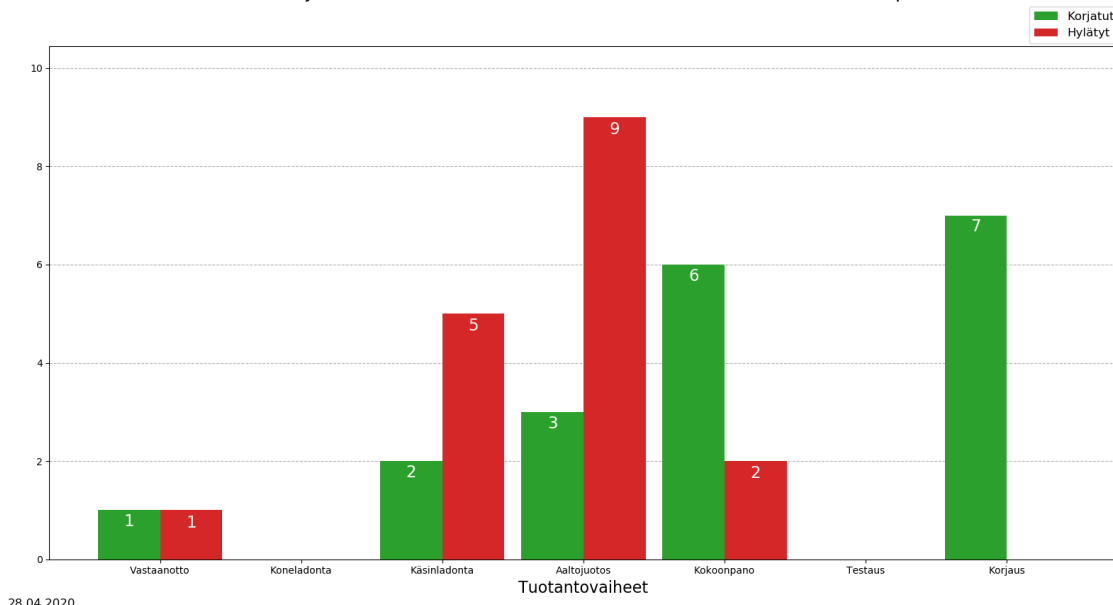


Kuva 11. Vikojen korjausajat laskettuna. Vika 2 on selvästi nopein vika korjata, joten niitä kannattaa ajatella korjattavan ensin.

Korjausajan analyysin (Kuva 11) menetelmät on selitetty erikseen luvussa 4.3.2. Analyysin voi suorittaa joko jollekin tietylle tuotteelle, jolloin saadaan juuri sille tuotteelle olevien vikojen korjausajat, tai kaikille tuotteille, jolloin saadaan vikojen yleinen korjausaika. Yleinen korjausaika voi olla hyödyllinen esimerkiksi uuden tuotteen vikojen korjausaikoja arvioidessa.

Esimerkki: Tuotteella 2 on ilmentynyt vikoja 1, 3 ja 4. Tuotteella 2 on ilmentynyt vikoja 2, 3, 4 ja 5. Näin ollen voidaan laskea yleiset korjausajat vioille 1-5. Viat 1, 2 ja 5 ovat suoraa joko tuotteen 1 tai 2 korjausajat, sillä niitä on ilmennyt vain toisessa, mutta vikojen 3 ja 4 korjausajat lasketaan nyt molemmille tuotteille kirjatusta ajoista.

Vian "Oikosulku" korjauksien onnistuneisuus eri tuotantovaiheissa viimeisen 60 päivän aikana



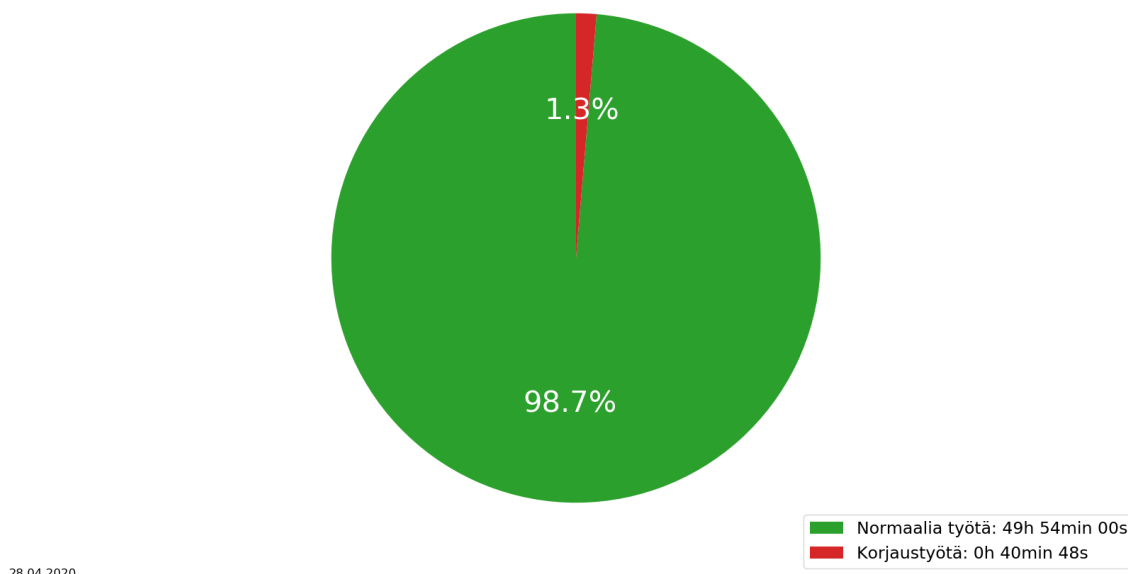
Kuva 12. Vikojen korjauksen onnistuneisuus eri työvaiheissa. Tämä vika ilmenee selvästi tuotannon keskivaiheilla, eikä sitä ole helppo korjata. Toisaalta korjausvaiheelle siirretyt viat on saatu aina korjattua.

Vikojen korjauksen onnistuneisuuden analyysia (Kuva 12) voidaan käyttää mittaamaan tietäntyyppisten virheiden määrää ja laatua. Sillä voi myös nähdä, missä tuotantovaiheessa virheet tapahtuvat tai missä niitä saadaan vähiten korjattua. Näin voidaan kohdistaa laadun parantamisen toimenpiteet oikein. Virheiden kehityksen saa ajoittamalla analyysin ja seuraamalla sen kehitystä. Ohjelmassa on myös vaihtoehto suorittaa analyysi vanhoille vikakoodeille. Vanhoille vikakoodeille ei suoriteta tuotantovaihejaottelua, joten tuloksena on vain ympyrädiagrammi kaikista kyseessä olevista virheistä.

Viimeinen analyysi on tuotteiden ajankäyttöä kuvaava analyysi (Kuva 13). Siinä lasketaan johonkin tuotteeseen käytetyn tuotantotyön ja korjaustyön suhde, jotta voidaan tarkastella tuotteeseen tehtyjen korjausten aiheuttamaa työajan menetystä. Kuvaajan selitteeseen on laskettu absoluuttiset työajat.

Analyysi voidaan suorittaa joko tietylle määrälle edellisiä eriä tai päiviä valitsemalla toiminnoksi joko *Ajankäyttö tuotteelle* tai *Ajankäyttö tuotteelle aikavälillä*. Ero on siinä, että päiviä käytettäessä rajaava tekijä on annettu määrä päiviä nykyisestä, kun taas eriä käytettäessä haetaan ensin päivävälillä olevat tuotantoerät ja sitten niille suoritettut korjauserät, jolloin laskusta jätetään pois korjauserät, joiden tuotantoerä on suoritettu ennen annettua aikaväliä.

Ajankäyttö tuotteelle Tuote2 viimeisen 60 päivän aikana



Kuva 13. Ajankäyttö tuotteelle viimeisen 60 päivän aikana. Korjaustyön osuus on hyvin pieni tuotantotyön osuuteen, joten kustannukset on minimoitu.

4.3.2 Vikojen korjausajan laskeminen

Vikojen korjausaikoja voidaan laskea tehtyjen korjauserien perusteella. Korjauserille kirjattut virheet ja niiden määrät, sekä korjauserälle käytetty työaika voidaan kirjata taulukkoon, josta yksittäisen vian korjausaika voidaan laskea matriisilaskuna, mikäli korjauseriä on tehty tarpeeksi monta. Jos tuotteelle on tehty m korjauseriä, joissa on kirjattu n erilaista vikaa, korjauserien määrän tulee olla vähintään yhtä suuri kuin vikojen määrä ($m \geq n$), jotta matriisi voidaan ratkaista.

Sanotaan esimerkiksi, että vikoja neljä: vika 1, vika 2, vika 3 ja vika 4. Niiden todelliset korjausajat ovat 12 min (0,2h), 30 min (0,5h), 60 min (1,0h) ja 90 min (1,5h). Korjauseriä on kahdeksan ja niillä on korjattu vikoja kuvan 14 mukaisesti.

	Vika 1	Vika 2	Vika 3	Vika 4	Erän aika
Korjauserä 1:	10	10	9	9	29.5
Korjauserä 2:	9	0	1	5	10.3
Korjauserä 3:	1	1	0	10	15.7
Korjauserä 4:	8	3	3	0	6.1
Korjauserä 5:	0	0	9	3	13.5
Korjauserä 6:	5	10	3	6	18.0
Korjauserä 7:	0	0	0	10	15.0
Korjauserä 8:	0	0	0	2	3.0

Kuva 14. Esimerkki virheiden ja korjausajan matriisista.

Erän ajat on kuvassa 14 laskettu todellisten korjausaikojen mukaisesti. Jos nämä ajat pitäisivät paikkansa, voitaisiin matriisista ottaa mitkä tahansa neljä riviä ja laskea niiden avulla korjausajat eri virheille. Todellisuudessa erän ajat saadaan erälle kirjatusta työtunneista, eivätkä niillä korjattujen vikojen korjausajat ole tietenkään samat. Voidaan ajatella, että jokaiselle vian korjaukselle on määriteltävissä jokin poikkeama todellisesta korjausajasta. Jos poikkeamat otetaan huomioon, ovat korjausaikojen arviot virheille erisuurat eri rivejä valittaessa. Joillakin kombinaatioilla rivejä tulosta ei välttämättä edes ole, mutta jos erilaisia toimivia kombinaatioita löytyy tarpeeksi useita, on mahdollista muodostaa jakauma korjausaikojen arvioista, missä todellinen korjausaika voidaan arvioida.

Keskeisen raja-arvolauseen perusteella voidaan sanoa, että korjausajat jakautuvat likimäärin normaalijakauman mukaisesti, sillä korjausaika itsessään koostuu pienemmistä eri korjausvaiheiden ajoista, joiden pituus vaihtelee lukuisien satunnaisien tekijöiden mukaan. [18.] Korjausajan poikkeama kuvastaa näiden satunnaisten tekijöiden summaa. Näin ollen korjausaikojen arvioista muodostettu jakauma jakautuu niin ikään normaalijakauman mukaisesti. Korjausaikojen arvioiden jakauman odotusarvo kertoo oletetun korjausajan, ja sen varianssi riippuu korjausaikojen poikkeamien mukaan.

Odotusarvo normaalijakaumaa jäljitteleville jakaumille voidaan laskea joko keskiarvon, mediaanin tai moodin avulla. Menetelmillä ilmenee kuitenkin eroja, mikäli jakaumat ovat johonkin suuntaan vinoja tai käytettävä aineisto on liian pieni. Laskutavaksi valittiin

mediaani, sillä se antaa parhaan arvion molemmat tekijät huomioon ottaen. Pienillä aineistoko'illa keskiarvo on altis virheille yksittäisien suurien poikkeamien kautta ja moodi taas jonkin arvion satunnaisesta enemmistöstä. Vinoutta voidaan pitää korjausajan arvion tendenssinä johonkin suuntaan, esimerkiksi korjaustöiden pitkittymistä kuvaavana tekijänä. Käyttämällä mediaania voidaan tämä tendenssi sisällyttää arvioon.

Yleistetään korjausajan arvion laskukaavan 1 mukaisesti. Keskeisen raja-arvolauseen kohtuullisen luottamusasteen saavuttaminen edellyttää tarpeeksi suurta arvojoukkoa. Arvojoukon koko on tässä $C(m, n)$, josta on vähennetty kombinaatiot, joilla korjausaikoja ei voida laskea. Ohjelmassa ratkaisun löytyminen ei edellytä tämän ehdon täyttymistä.

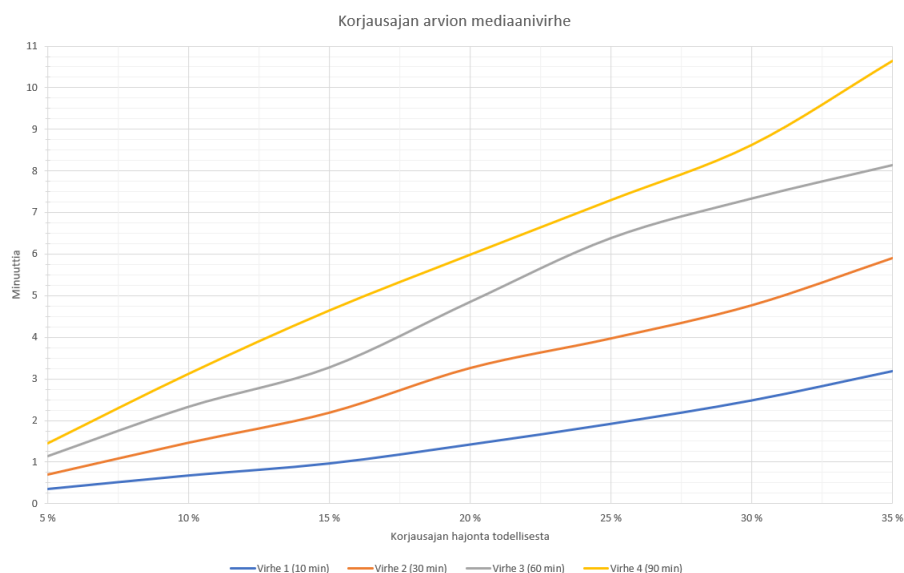
$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \text{median} \left(\text{Combination} \left(\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{pmatrix}^{-1} \begin{pmatrix} \Delta_{11} & \Delta_{12} & \Delta_{13} & \dots & \Delta_{1n} \\ \Delta_{21} & \Delta_{22} & \Delta_{23} & \dots & \Delta_{2n} \\ \Delta_{31} & \Delta_{32} & \Delta_{33} & \dots & \Delta_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta_{m1} & \Delta_{m2} & \Delta_{m3} & \dots & \Delta_{mn} \end{pmatrix}^{-1} \right), n \right) \quad (1)$$

$n = \text{number of unique faults}, \quad m = \text{number of batches},$
 $x = \text{fix time for fault } (n), \quad b = \text{time to fix all faults in a batch } (m)$
 $a = \text{amount of faults for a fault } (n) \text{ in a batch } (m),$
 $\Delta = \text{fixtime delta for a fault } (n) \text{ in a batch } (m)$

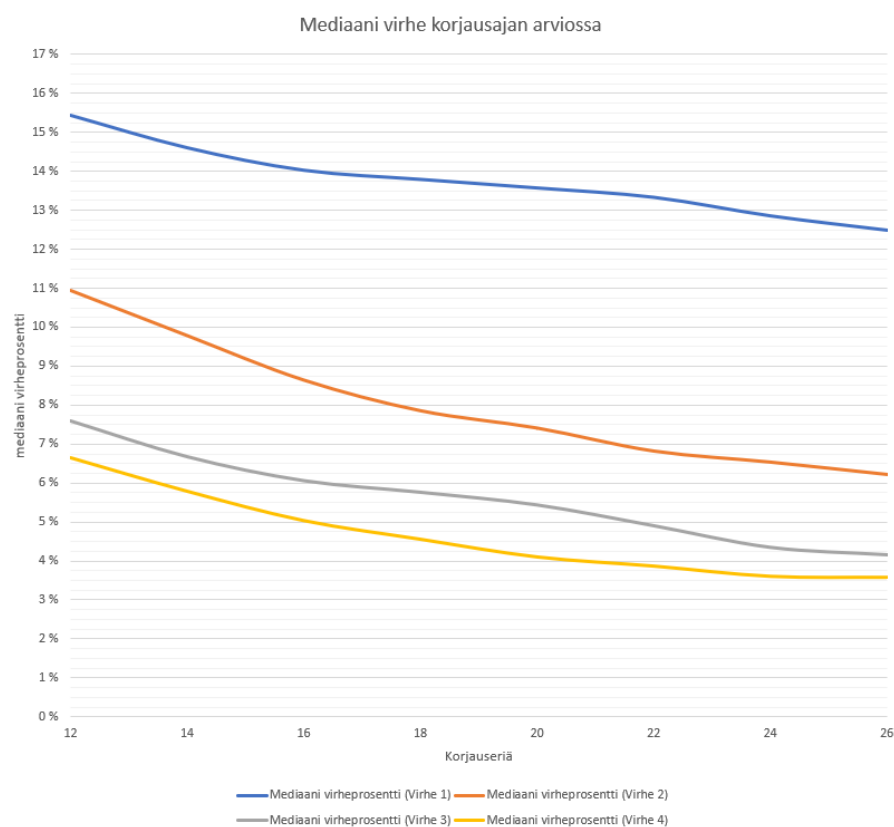
4.3.3 Korjausajan arvion analysointi

Kuvassa 15 on kuvattu, kuinka korjausajan arvion virhe todellisesta kasvaa mitä suurempi hajonta korjausajalla eri korjausten välillä on. Arvion virhe on laskettu mediaanina 500 arvioinnin sarjasta jokaiselle x-akselin maksimipoikkeamalle, jossa jokaisessa arvio tehtiin laskemalla mediaani 12 erän virheistä.

Maksimipoikkeama tarkoittaa tässä suurinta prosentuaalista poikkeamaa todellisesta korjausajasta (\pm), minkä jonkin erän virheen korjausaika saattoi kyseessä olevissa 500 arvioinnissa saada. Kaavassa 1 tämä on suurin mahdollinen arvo muuttujalle Δ . Näin voidaan havainnollistaa useita eri 12 korjauserän skenaarioita ja niissä todennäköisesti tapahtuvaa korjausajan arvion virhettä. On huomioitava, että erien lukumäärää kasvatessa arvion virhe pienenee. Tätä on havainnollistettu kuvassa 16, jossa samoilla periaatteilla tehty jokaiselle erämäärälle $\pm 20\%$ maksimipoikkeamalla 500 analyysiä jokaista erämäärää kohden.



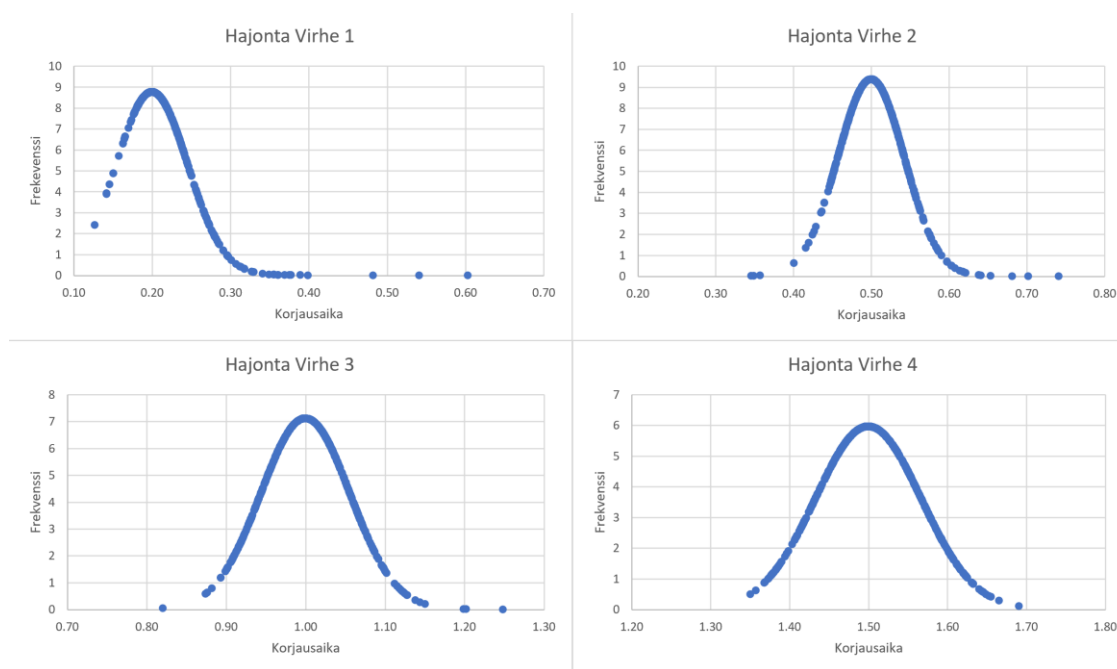
Kuva 15. Korjausajan arvion mediaanivirhe. Virhe kasvaa nopeammin pidemmille korjausajoille.



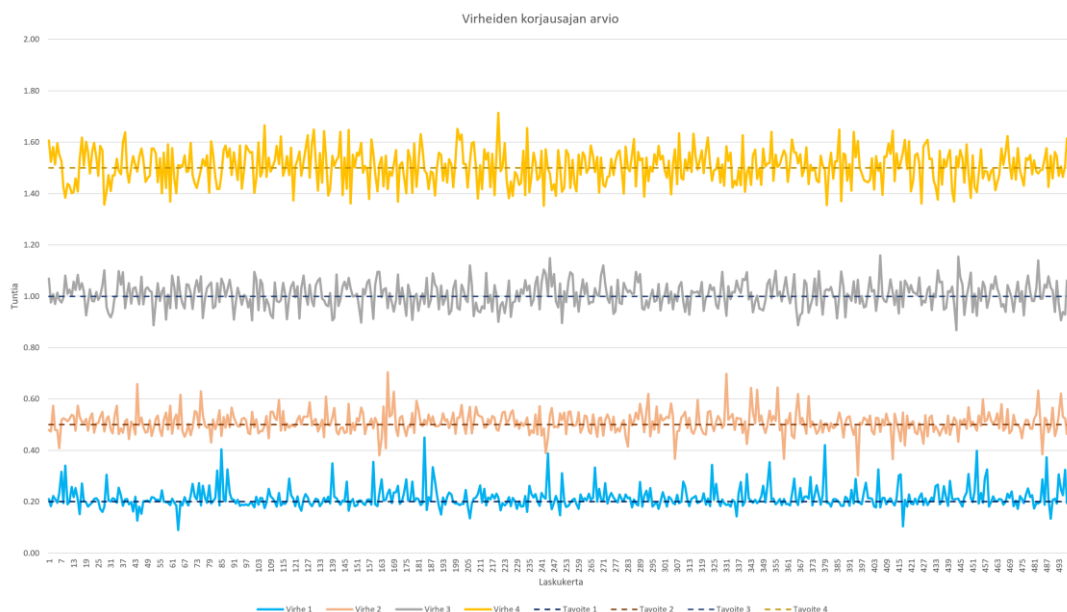
Kuva 16. Mediaanivirheprosentin kehitys eri määrillä eriä. Virheprosentit laskevat virheen ajasta huolimatta samaa tahtia.

Arvion onnistuneisuutta voidaan myös havainnollistaa piirtämällä edellä mainitut 500 korjausajan arviointikertaa joltakin korjausajan maksimipoikkeamalta ja erämäärältä. Kuvassa 17 arviot on sovitettu normaalijakaumalle, ja niissä on käytetty 12 erää ja $\pm 10\%$ maksimipoikkeamaa. Kuvassa 18 saadut arvot on esitetty viivakaaviossa sellaisenaan.

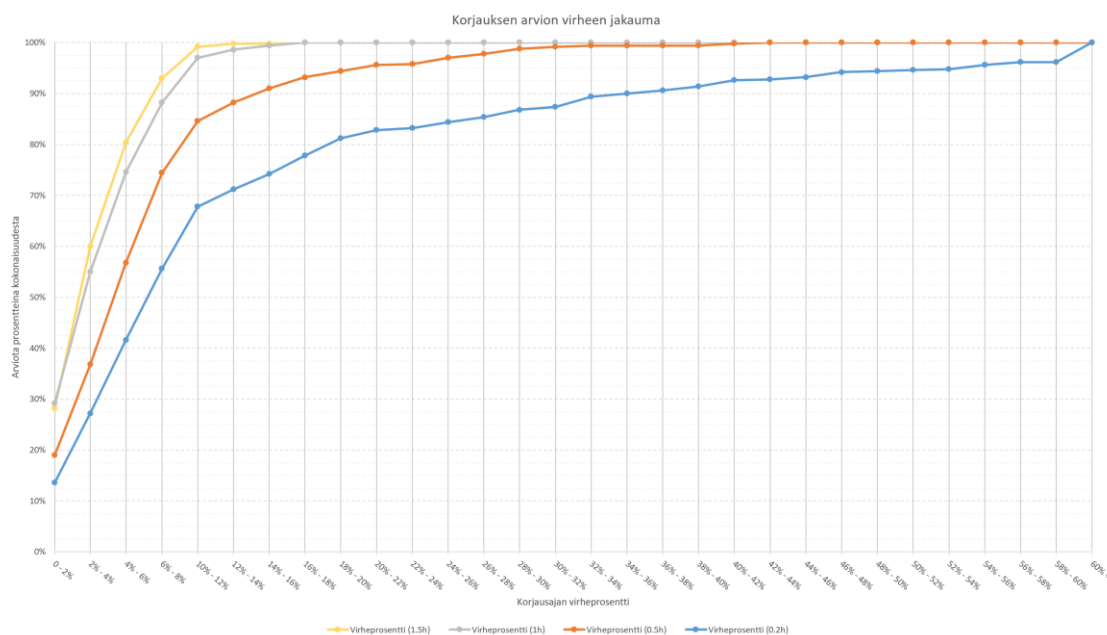
Molemmista voidaan todeta, että noin 90 % tehdyistä arvioinneista poikkeaa vain 0,1h todellisesta arvosta, riippumatta virheen korjausajasta. Kuvassa 19 tämä voidaan todeta kumulatiivisten prosenttiosuuksien kautta, ja kuvasta 20 voidaan tarkastella tarkemmin eri virheprosenttiosuuksien kertymiä histogrammin kautta.



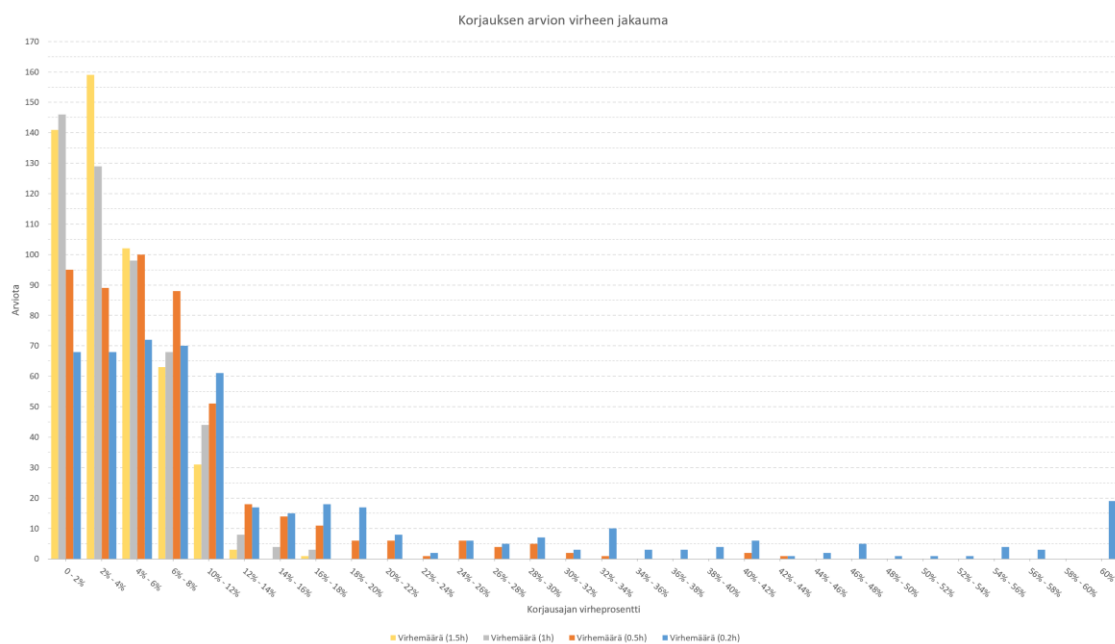
Kuva 17. Eri virheiden korjausajan arvion hajonta. Hajonta virheelle 1 vääristyy hieman eteenpäin, koska negatiivisia vastauksia ei hyväksytty. Kaikista hajonnoista voidaan kuitenkin todeta selvä normaalijakauma.



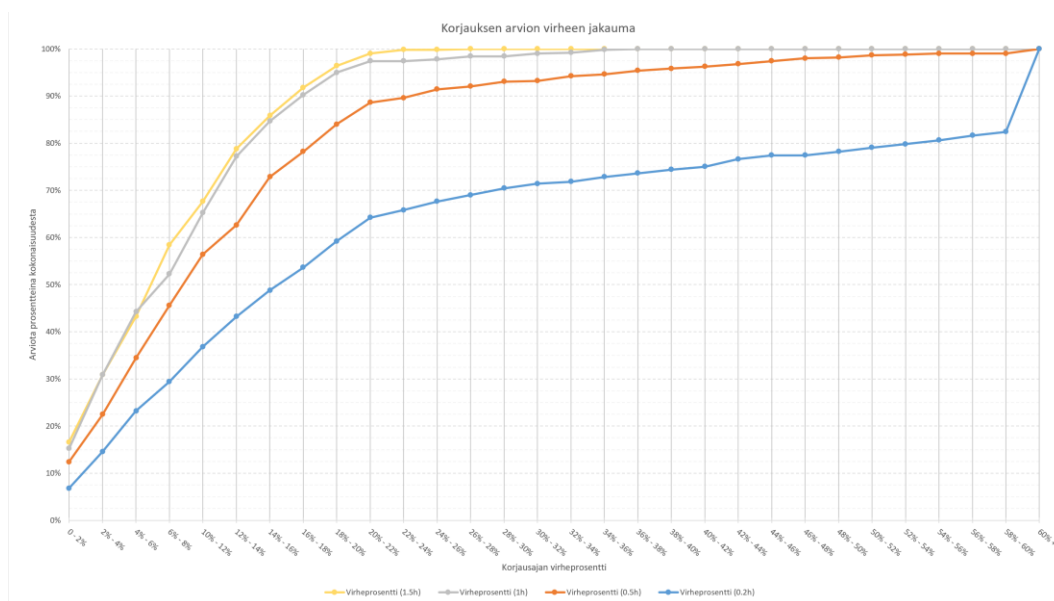
Kuva 18. Virheiden korjausajan arvio 500 eri laskentakerran aikana. 12 erää, $\pm 10\%$ maksimipoikkeama, 500 arviota. Eri virheistä heittelevät suunnilleen yhtä paljon.



Kuva 19. Virheiden korjausajan arvion prosentuaalisen virheen kumulatiivinen jakauma eri korjausajoille. 12 erää, $\pm 10\%$ maksimipoikkeama, 500 arviota. Voidaan todeta kuvaa 18 paremmin, että arvion virhe kasautuu jokaiselle virheelle suunnilleen samassa suhteessa, tosin pienemmällä korjausajalla hieman hitaammin.

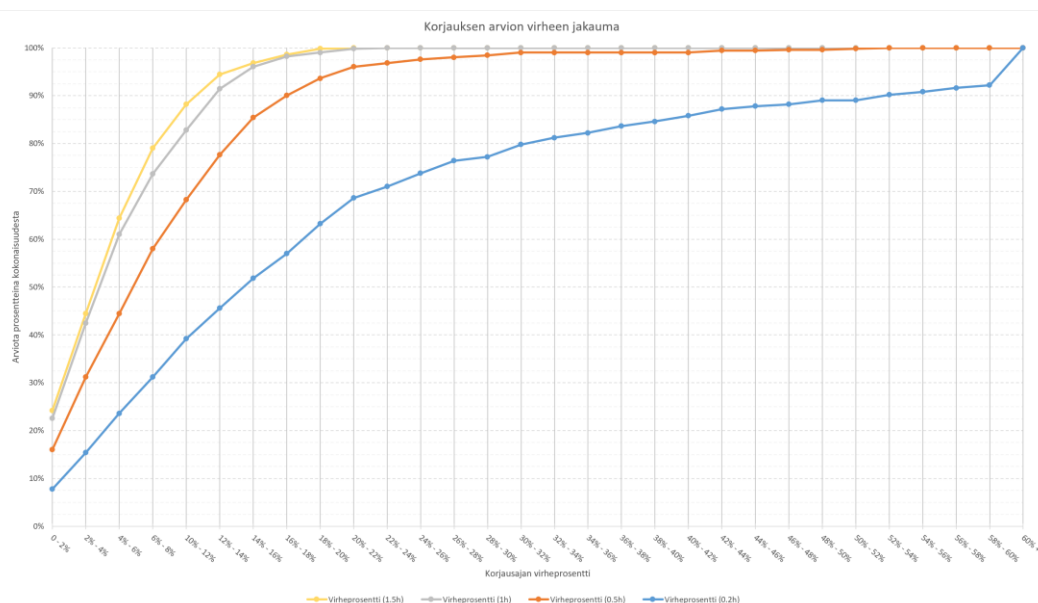


Kuva 20. Virheen korjausajan arvion prosentuaalisen virheen histogrammi eri korjausajoille. 12 erää, $\pm 10\%$ maksimipoikkeama, 500 arviota. Sama kuin kuvassa 19, mutta inkrementaalaisesti kuvattuna.



Kuva 21. Virheiden korjausajan arvion prosentuaalisen virheen kumulatiivinen jakauma eri korjausajoille. 12 erää, $\pm 20\%$ maksimipoikkeama, 500 arviota. Suuremman maksimipoikkeaman takia virheprosentit kasautuvat hitaammin, mutta lukuun ottamatta pienintä virheaikaa silti hyvin nopeasti.

Kuvassa 21 korjausajan virheen maksimipoikkeamaa on nostettu ± 20 %:iin, jolloin korjausajan arvion virheen 90 % alue kasvaa, erityisesti pienillä korjausajoilla. Kuvassa 22 erämäärää on nostettu, mikä pienentää 90 % aluetta, tarkentaen arviota.



Kuva 22. Virheiden korjausajan arvion prosentuaalisen virheen kumulatiivinen jakauma eri korjausajoille. 24 erää, ± 20 % maksimipoikkeama, 500 arviota. Erämäärän kasvun takia virheprosentit kasautuvat nopeammin, mutta eivät vielä yhtä nopeasti kuin aiemmin.

Laskentamenetelmistä tehtyjen analyysien perusteella voidaan sanoa, että hyvän analyysin saamiseksi on kolme sääntöä:

1. Keskeisen raja-arvoperiaatteen ylläpitämiseksi tulee valideja laskukombinaatioita olla "tarpeeksi".
2. Mitä pienempi vian korjausajan maksimipoikkeama (Δ) on, sitä parempi arvio.
3. Mitä enemmän korjauseriä on suhteessa erilaisiin vikoihin, sitä parempi arvio.

Algoritmia arvioidessa on hyvä huomata, että virheiden laskuaika kasvaa nopeasti, mitä enemmän erilaisia virheitä ja korjauseriä on. Matriisien ratkaisuun käytetty NumPy:n lineaaristen yhtälöryhmien ratkaisumenetelmän aikavaatimus on $O(n^3)$ [19.], ja koska se suoritetaan kaikille $C(n, k)$ kombinaatioille, on yhteinen aikavaatimus $O(n^4)$. Kombinaatioiden aikavaatimus on $O(C(n, k))$. Suorituskyvyn takaamiseksi algoritmiin voidaan las- kutoimitusten kasvaessa lisätä esimerkiksi raja- aikavälin kautta, jotta laskut saadaan

suoritettua tarpeeksi nopeasti. Korjausaika-analyysi on kuitenkin ajateltu suoritettavan vain aika ajoin, jolloin analyysin tarkkuus on suoritusnopeutta tärkeämpää.

5 Yhteenveto

Opinnäytetyön tarkoituksena oli uudistaa yrityksen laadunhallintajärjestelmiä, sekä selkeyttää toimintaa laatua käsittelevissä toimissa. Tätä varten asetettiin neljä tavoitetta: laadudatan keräämisen standardoiminen, tuotannolle paremman kuvan antaminen laadun tilanteesta, korjaustoimenpiteiden priorisointi ja toimenpiteiden vaikutuksien analysointi. Laadudatan kerääminen standardoitiin uudella virhekoodijärjestelmällä ja muiden tavoitteiden ratkaisemiseksi kehitettiin ohjelmisto, jolla tuotannon laadudataa voi analysoida.

Laadunvalvonnan kehittäminen on pitkä ja monitasoinen prosessi. Alussa projektin mitakaava oli huomattavasti laajempi, mutta sitä jouduttiin rajaamaan, jotta tuloksia saataisiin suotuisalla aikavälillä. Käytössä olevat järjestelmät ja standardit asettavat useita eri vaatimuksia ja rajoitteita, joka takia työn aikana on käyty läpi useita eri vaihtoehtoja ennen päätymistä nykyisiin uudistuksiin. Palautetta on otettu vastaan niin tuotannonsuunnittelusta kuin itse tuotannosta, ja työn lopputulosta on yritetty muokata heidän toiveitensa mukaisesti.

Työn tavoitteet ovat toteutuneet mielestäni hyvin. Ratkaisut ovat joustavia, eivätkä ne aiheuta ylimääräistä rasitetta niin tuotannolle kuin sen suunnittelullekaan. Uudet menetelmät selkeyttävät laadun ylläpitoa sekä erityisesti helpottavat sen tulkitsemista. Menetelmiä on jatkossa myös helppo muuttaa, sillä perustavanlaatuinen työ esimerkiksi kerätyn laadudatan analysoimiseen on jo tehty. Uskonkin, että muutoksia tehdään sitä mukaa kun uusi laadudatankeräysjärjestelmä tulee käyttöön eri yrityksen valmistamille tuotteille.

Seuraava askel järjestelmän käyttöönotossa on kouluttaa työntekijät uuteen järjestelmään, sekä suunnitella sen implementointi jokapäiväiseen työnkulkuun. Jotelin kanssa olemmeideoineet, että automaattiset analyysit voitaisiin saada tuotantoon näkyviin esimerkiksi televisioruudulle, johon taulukot automaattisesti päivittyisivät. Ensisijaisen tärkeää on kuitenkin huolehtia siitä, että työntekijät käyttävät järjestelmää oikein, jotta

kerätty data voidaan hyödyntää. Tämä tarkoittaa sitä, että jokaiselle on oltava selvää, miten missäkin vikatilanteessa toimitaan, ja mitä milläkin vikakoodilla tarkoitetaan. Työntekijöiden on myös huolehdittava tyyppivikojen ylläpidosta, ja siitä että ne todellakin ovat tarpeellisia. Tuotannonsuunnittelun taas on seurattava laadun kehitystä uusien vikakoodien myötä, sekä puututtava ilmeneviin epäkohtiin tai puutoksiin. Heidän on myös opittava käyttämään laitudatan analysointia varten tehtyä ohjelmistoa, sekä annettava palautetta sen toiminnasta. Näin laadunhallintaa ja uusia menetelmiä voidaan kehittää yhä eteenpäin.

Lähteet

- 1 Jotel Oy -verkkosivut. Verkkoaineisto. <<https://jotel.com/>>. Luettu 22.5.2020.
- 2 Asiakastieto, Jotel Oy. 2020. Verkkoaineisto. Asiakastieto <<https://www.asiakastieto.fi/yritykset/fi/jotel-oy/08298307/yleiskuva>>. Luettu 25.5.2020.
- 3 Python (Ohjelmointikieli). Verkkoaineisto. <[https://fi.wikipedia.org/wiki/Python_\(ohjelmointikieli\)](https://fi.wikipedia.org/wiki/Python_(ohjelmointikieli))>. 19.4.2020. Luettu 22.5.2020.
- 4 About pandas. 2020. Verkkoaineisto. Pandas. <<https://pandas.pydata.org/about/index.html>>. Luettu 22.5.2020.
- 5 NumPy -verkkosivut. 2020. Verkkoaineisto. Numpy. <<https://numpy.org/>>. Luettu 22.5.2020
- 6 Matplotlib: Visualization with Python. Verkkoaineisto. Matplotlib. <<https://matplotlib.org/>>. Luettu 22.5.2020.
- 7 The SciPy Stack specification, 2020. Verkkoaineisto. SciPy. <<https://www.scipy.org/stackspec.html>>. Luettu 22.5.2020
- 8 Graphical User Interfaces with Tk. 2001. Verkkoaineisto. Python Software Foundation. <<https://docs.python.org/3/library/tk.html>>. Päivitetty 25.5.2020. Luettu 25.5.2020.
- 9 Natsort. 2020. Verkkoaineisto. Python Software Foundation. <<https://pypi.org/project/natsort/>>. Luettu 22.5.2020.
- 10 Composable cycles. 2016. Verkkoaineisto. Python Software Foundation. <<https://pypi.org/project/natsort/>>. 16.2.2016. Luettu 22.5.2020.
- 11 Wisetime järjestelmä. 2020. Verkkoaineisto. Wisetime Oy. <<https://www.wisetime.fi/jarjestema.html>>. Luettu 22.5.2020.
- 12 Create Custom Color Palettes. 2003 Verkkoaineisto. <https://help.tableau.com/current/pro/desktop/en-us/formatting_create_custom_colors.htm>. Luettu 22.5.2020.
- 13 IPC International. 2017. Acceptability of Electronic Assemblies. Saatavilla: <<https://shop.ipc.org/IPC-A610G-English-D>>. Luettu 22.5.2020.

- 14 Euroopan parlamentti ja neuvosto. 2014. Direktiivi 2014/34/EU. Luettavissa: <<https://eur-lex.europa.eu/legal-content/FI/TXT/HTML/?uri=CELEX:32014L0034>>. Päivitetty 26.2.2014. Luettu 22.5.2020.
- 15 Herres, David. 2015. The meaning of UL and CSA marks. Verkkoaineisto. Design World. <<https://www.designworldonline.com/the-meaning-of-ul-and-csa-marks/>>. Päivitetty 14.12.2015. Luettu 22.5.2020.
- 16 Stallcup, James. 2001. Installing Custom-Made Equipment Correctly. Verkkoaineisto. Endeavor Business Media, LLC. <<https://www.ecmweb.com/national-electrical-code/code-basics/article/20890826/installing-custommade-equipment-correctly>>. Päivitetty 1.2.2020. Luettu 22.5.2020.
- 17 SFS ry, 2015 FS-EN ISO 9001:2015. Saatavilla: <<https://sales.sfs.fi/fi/index/tuotteet/SFS/CENISO/ID2/9/394310.html.stx>>. Päivitetty: 23.10.2015. Luettu 22.5.2020.
- 18 Central Limit Theorem, 2010. Verkkoaineisto. Encyclopedia of Mathematics. <https://encyclopediaofmath.org/wiki/Central_limit_theorem>. Päivitetty 11.5.2012. Luettu 22.5.2020.
- 19 Blackford, Susan. 1999. LAPACK Benchmark. Verkkoaineisto. Netlib. <<http://www.netlib.org/lapack/lug/node71.html>>. Päivitetty 01.10.1999. Luettu 22.5.2020.